



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM  
INSTITUTO DE COMPUTAÇÃO - ICOMP  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI



# **Reconhecimento de Atividades Humanas baseado na Análise de Fluxo Contínuo de Dados Simbólicos**

Wesllen Sousa Lima

Manaus - Amazonas  
5 de Abril de 2019

Wesllen Sousa Lima

# **Reconhecimento de Atividades Humanas baseado na Análise de Fluxo Contínuo de Dados Simbólicos**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito final para aprovação da proposta.

Área de concentração: Aprendizagem de máquina.

Orientador: Prof. Dr. Eduardo James Pereira Souto

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

L732r Lima, Wesllen Sousa  
Reconhecimento de Atividades Humanas baseado na Análise de Fluxo Contínuo de Dados Simbólicos / Wesllen Sousa Lima. 2019  
162 f.: il. color; 31 cm.

Orientador: Eduardo James Pereira Souto  
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Reconhecimento de Atividades Humanas. 2. Sensores Inerciais. 3. Algoritmos de Representação Simbólica. 4. Data Streaming. I. Souto, Eduardo James Pereira II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
INSTITUTO DE COMPUTAÇÃO



UFAM

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

## FOLHA DE APROVAÇÃO

**"Reconhecimento de atividades humanas baseada na análise de  
fluxo contínuo de dados simbólicos"**

**WESLLEN SOUSA LIMA**

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Eduardo James Pereira Souto - PRESIDENTE

Prof. Marco Antonio Pinheiro de Cristo - MEMBRO INTERNO

Prof. Raimundo da Silva Barreto - MEMBRO INTERNO

Prof. Antonio Alfredo Ferreira Loureiro - MEMBRO EXTERNO

Prof. João Gama - MEMBRO EXTERNO

Manaus, 05 de Abril de 2019

Dedico este trabalho aos meus pais  
Antonio Melo e Eliacy Barbosa  
pelo incentivo e apoio incondicional  
que permitiram chegar até aqui.

# Agradecimentos

À Deus.

Aos meus pais Antonio Melo e Eliacy Barbosa pelos esforços, mesmo nos momentos em que não há condições para isso. Sou grato pelas orações, pelos conselhos e pelos incentivos. Sem ajuda de vocês essa conquista não seria possível. Amo vocês!

Ao meu orientador, Eduardo Souto. Obrigado por ter me aceitado como seu aluno e por ter acreditado em mim. Obrigado por ter contribuído com a minha formação, serei eternamente grato!

À minha noiva Thalita Lucena pela paciência e apoio durante esses 4 anos de relacionamento.

Agradeço a todos que contribuíram direta ou indiretamente para realização deste trabalho. Em especial, as amigas que foram construídas durante essa empreitada.

Agradeço ao programa de pós-graduação do Instituto de computação (IComp) na Universidade Federal do Amazonas (UFAM) pela oportunidade e suporte que tem me dado. Também agradeço aos integrantes do grupo de pesquisa *Emergency Technologies and Security Systems* (ETSS) os quais conviveram comigo durante o período da minha formação. Obrigado!

À FAPEAM pelo apoio financeiro determinante para a conclusão deste trabalho. Além disso, agradeço às instituições que também contribuíram para o desenvolvimento desta tese, UOIT (University of Ontario Institute of Technology), representadas pelos professores Richard Pazzi e Khalil El-Khatib e INESC TEC representado pelo professor João Gama.

*Tudo tem o seu tempo determinado,  
e há tempo para todo o propósito debaixo do céu.*

*Eclesiastes 3:1*

# Resumo

A capacidade de sensoriamento dos smartphones tem permitido o desenvolvimento de soluções para reconhecimento de atividades humanas (do inglês, *Human Activity Recognition* - HAR) com o propósito de entender melhor o comportamento humano por meio de técnicas computacionais. No entanto, tais soluções têm tido dificuldades para atuar em cenários dinâmicos por não observarem a evolução dos dados ao longo do tempo e pelo alto consumo de recursos computacionais, como memória, processamento e energia das baterias. Isso ocorre porque o problema de HAR para smartphones têm sido resolvidos por meio de modelos de classificação gerados por algoritmos de aprendizagem de máquina *offline* que, nesse caso, são limitados por um histórico de dados com poucas informações sobre às atividades humanas. O problema dessa abordagem é que as atividades humanas mudam constantemente ao longo do tempo e são fortemente influenciadas pelo ambiente físico e pelo perfil dos usuários.

Para superar esses problemas esta tese de doutorado propõe uma nova abordagem para reconhecimento de atividades humanas baseada na análise do fluxo contínuo de dados simbólicos. Nossa abordagem permite o desenvolvimento de sistemas HAR de baixo custo, escaláveis e capazes de se adaptarem a mudanças ao longo do tempo. Neste contexto, esta tese propõe um *framework*, denominado DISTAR (*DI*crete *ST*ream learning for *ACT*ivity *RE*cognition), responsável por padronizar o processo de análise do fluxo contínuo de dados e geração de modelos adaptativos que observam a evolução dos dados ao longo do tempo sem a necessidade de armazenar um histórico de dados. O *framework* DISTAR utiliza como base os algoritmos de representação simbólica que são conhecidos na literatura por reduzirem a dimensionalidade e a numerosidade dos dados. Além disso, esta tese também propõe um novo algoritmo *online* adaptativo, denominado NOHAR (*NO*velty *DI*crete *DA*ta *ST*ream for *HA*uman *ACT*ivity *RE*cognition), que utiliza como base o *framework* DISTAR. Os resultados experimentais usando três bases de dados mostram que o NOHAR é mais rápido, em média, 13 vezes comparado ao estado da arte e é capaz reduzir o consumo de memória, em média, 99,97%. Em termos de precisão, o NOHAR é similar ao estado da arte.

A principal contribuição desta tese consiste na abertura de um novo ramo de pesquisa na área de HAR o qual utiliza dados simbólicos para o desenvolvimento de soluções de baixo custo, escaláveis e adaptativas. Além disso, o *framework* DISTAR e o algoritmo adaptativo NOHAR são propostas inéditas para a área de processamento de fluxo contínuo de dados (do inglês, *Data Streaming*) que podem ser exploradas em outros domínios e problemas.

**Palavras chaves:** reconhecimento de atividades humanas; sensores inerciais; algoritmos de representação simbólica; data streaming;



# Abstract

The smartphones sensing capabilities have enabled the development of Human Activity Recognition (HAR) solutions for better understanding human behavior through computational techniques. However, these solutions have been difficult to perform in dynamic scenarios because they do not observe data evolution over time and the high consumption of computational resources, such as memory, processing and energy. This occurs because the HAR problem for smartphones has been solved through classification models generated by offline machine learning algorithms that, in this case, are limited by a data history with little information about human activities. The problem with this approach is that human activities change constantly over time and are strongly influenced by the physical environment and the user's profile.

To overcome these problems this doctoral thesis proposes a new approach to recognize human activities based on the symbolic data streaming analysis. Our approach enables the development of low-cost, scalable HAR systems capable of adapting to data change over time. In this context, this thesis proposes a framework called DISTAR (DIScrete STREAM learning for Activity Recognition), responsible for standardizing the analysis of data stream process and generation of adaptive models that observe the data evolution over time without storing a data history. The DISTAR framework uses the symbolic representation algorithms known for reducing the dimensionality and numerosity of the data. In addition, this thesis also proposes a new adaptive online algorithm, called NOHAR (NOvelty discrete data stream for Human Activity Recognition), which uses as basis the DISTAR framework. Experimental results using three databases show that NOHAR is 13 times faster compared to the state of the art and is able to reduce memory consumption by an average of 99.97%. In terms of accuracy, NOHAR is similar to the state of the art.

The main contribution of this thesis is the opening of a new research branch in the HAR area which uses symbolic data for the development of low cost, scalable and adaptive solutions. In addition, the DISTAR framework and the adaptive algorithm NOHAR are unprecedented proposals for the data streaming area that can be explored in other domains and problems.

**Keywords:** recognition of human activities; inertial sensors; algorithms of symbolic representation; data streaming;

# Sumário

Capítulo 1 Introdução.....	18
1.1. Problema .....	19
1.2. Hipótese .....	20
1.3. Objetivo .....	22
1.4. Contribuições .....	23
1.5. Organização do Documento.....	24
Capítulo 2 Reconhecimento de Atividades Humanas .....	26
2.1. Entendendo as Atividades Humanas .....	26
2.2. Entendendo a Evolução da Área de HAR.....	27
2.2.1 Contexto Histórico.....	27
2.2.2 Eventos Históricos baseados nos Smartphones .....	28
2.3. Sensores nos Smartphones.....	29
2.3.1 Acelerômetro.....	30
2.3.2 Giroscópio.....	31
2.3.3 Magnetômetro .....	31
2.4. Abordagens utilizadas para reconhecer as Atividades Humanas .....	31
2.5. Desafios e Oportunidades .....	33
2.6. Aplicações Práticas.....	34
2.7. Considerações Finais .....	35
Capítulo 3 Metodologia padrão de HAR e Trabalhos Relacionados .....	37
3.1. Metodologia Padrão para Análise de Dados .....	37
3.2. Coleta de Dados .....	38
3.2.1 Bases de Dados.....	41
3.3. Segmentação .....	42
3.4. O que são características?.....	43
3.4.1 Domínio do Tempo.....	44
3.4.2 Domínio da Frequência.....	46
3.5. Extração de Características .....	47
3.5.1 Redução de Dimensionalidade dos Dados .....	49
3.6. Treinamento e Classificação .....	50
3.6.1 Algoritmos de Aprendizagem Rasos .....	51
3.6.2 Algoritmos de Aprendizagem Profundos.....	51

3.6.3 Métricas de Avaliação .....	53
3.7. Fusão de Dados .....	54
3.8 Considerações Finais .....	55
Capítulo 4 Representação Simbólica dos Dados .....	56
4.1. Características do Domínio Discreto .....	56
4.1.1 Extração de Características .....	57
4.1.2 Discretizadores .....	58
4.2. Vantagens do Domínio Discreto no contexto HAR baseado em Smartphones .....	64
4.3. Algoritmos de Aprendizagem para o Domínio Discreto .....	65
4.3.1 Abordagens baseadas em Modelos Vetoriais .....	66
4.3.2 Abordagens baseadas em Algoritmos Rasos .....	69
4.4. Séries Temporais Multidimensionais .....	72
4.5. Trabalhos Relacionados .....	74
4.6. Considerações Finais .....	74
Capítulo 5 Aprendizagem Online .....	76
5.1. Online HAR .....	76
5.2. Conceitos de Aprendizagem Online .....	77
5.3. Aprendizagem Online baseada em Detecção de Novidades .....	79
5.4. Baselines .....	80
5.4.1 Aprendizagem Estacionária .....	81
5.4.2 Evolução de Dados .....	82
5.4.3 Detecção de Novidades .....	84
5.5. Considerações Finais .....	86
Capítulo 6 Método Online Simbólico para HAR .....	87
6.1. Fundamentos da Proposta .....	87
6.2. Framework DISTAR .....	88
6.3. Algoritmo NOHAR .....	90
6.3.1 Discretização .....	93
6.3.2 Geração dos Histogramas de Palavras .....	95
6.3.3 Aprendizagem .....	97
6.4. Considerações Finais .....	100
Capítulo 7 Experimentos e Resultados .....	101
7.1. Protocolo Experimental .....	101
7.1.1 Métricas .....	102
7.1.2 Bases de dados utilizados nos experimentos .....	102
7.2. Avaliação dos Algoritmos Rasos e Profundos .....	103

7.3. Avaliação dos Algoritmos de Representação Simbólica .....	104
7.3.1 Baselines .....	104
7.3.2 Configurações e Parâmetros .....	105
7.3.3 Resultados .....	105
7.4. Avaliação do NOHAR .....	109
7.4.1 Baselines .....	110
7.4.2 Configurações e Parâmetros .....	110
7.4.3 Resultados .....	111
7.5 Considerações Finais .....	122
Capítulo 8 Conclusões e Trabalhos Futuros .....	124
8.1 Trabalhos futuros .....	126
8.2 Publicações .....	126
Bibliografia.....	128
Apêndice A.    Bases de Dados .....	144
1.1. UCI-HAR.....	144
1.2. Shoaib .....	145
1.3 WISDM .....	145
Apêndice B.    Métodos e Características usados em Sistemas HAR .....	147
Apêndice C.    Avaliação dos Algoritmos Offline .....	149
1.1. Protocolo Experimental .....	149
1.2. Domínio do Tempo e Frequência.....	149
1.2.1 - 1º Cenário - Características dos domínios de Tempo e Frequência .....	150
1.2.2 - 2º Cenário: Técnicas de Redução de Ruídos dos Sinais .....	154
1.2.3 - 3º Cenário: Redução de Dimensionalidade.....	155
1.2.4 - 4º Cenário: Janela de Tempo, Sobreposição e Frequência .....	156
1.2.5 - 5º Cenário: Outras Bases de Dados .....	159
1.3. Fusão de Dados .....	159
1.4. Posição do Smartphone no Corpo do Usuário .....	161
1.5. Aprendizagem Profunda .....	162
Apêndice D.    Ferramenta BHAR-Benchmarking HAR.....	163

# Lista de Figuras

Figura 2.1 Exemplos de dados extraídos de um acelerômetro com padrões de diferentes atividades executadas ao longo do tempo. Fonte [96].	30
Figura 2.2 Exemplo das rotações em cada eixo do smartphone. Fonte: Google.	31
Figura 2.3 Tipos de abordagens empregada nas soluções de HAR. Fonte: autor.	32
Figura 3.1 Metodologia geral para soluções de HAR. Fonte: autor.	38
Figura 3.2 Exemplo de coleta e armazenamento de dados dos sensores inerciais. Fonte: autor.	39
Figura 3.3 Exemplo de segmentação não-sobreposta (a) e segmentação com sobreposição (b). Fonte [93].	42
Figura 3.4 Exemplo de extração de características e geração de uma base de dados secundária. Fonte: autor.	48
Figura 3.5 Conjunto de etapas baseadas na extração de características automáticas por meio da utilização de algoritmos de aprendizagem profunda. Fonte: autor.	52
Figura 4.1 Exemplo de uma série temporal discretizada. Adaptado de [194].	56
Figura 4.2 Exemplos visuais dos resultados dos principais algoritmos de discretização identificados na literatura. Fonte [194].	57
Figura 4.3 Exemplo da diferença entre seqüência e subsequência. Fonte [95].	58
Figura 4.4 Tabela estatística utilizada na definição dos breakpoints. Fonte [48].	59
Figura 4.5 Exemplo de discretização de uma série temporal com o SAX. Fonte: autor.	59
Figura 4.6 Exemplo visual de uma discretização de uma subsequência com 3 resoluções do algoritmo iSAX. Fonte: autor.	60
Figura 4.7 Exemplo visual da diferença entre o SAX e o SFA. Fonte [194].	61
Figura 4.8 A subsequência T é aproximada utilizando o DFT e quantizada utilizando o MCB para, finalmente, gerar a palavra DAAC de tamanho 4 e alfabeto de tamanho 6. Fonte [194].	61
Figura 4.9 Exemplo de geração do MCB por meio dos coeficientes do MFT. Fonte [194].	62
Figura 4.10 Exemplo visual da distribuição dos bins e dos breakpoints em um MCB. Neste caso, o tamanho da palavra é 4 e do alfabeto é 6. Fonte [194].	63
Figura 4.11 Resumo de todas as etapas de processamento do SFA. Adaptado de [194].	63
Figura 4.12 Exemplo de redução de dimensionalidade e numerosidade em uma série temporal discretizada e representadas por palavras. Neste caso, os números indicam os índices, as palavras em negrito indicam as palavras repetidas.	64
Figura 4.13 Exemplo de um histograma.	66
Figura 4.14 Exemplo das etapas de processamento de dados do algoritmo SAX-VSM. Fonte [40].	68
Figura 4.15 Exemplo de discretização e geração do histograma pelo SFA. Fonte [42].	69
Figura 4.16 Exemplo de dois histogramas BOSS.	70
Figura 4.17 Diferença entre os histogramas gerados pelo SFA (esquerda) e SFA-W (direita). Fonte [43].	70
Figura 4.18 Exemplo de como funciona o processo de discretização do SFA-W. Fonte [43].	71
Figura 4.19 Empilhamentos das palavras de diferentes coordenadas.	73
Figura 5.1 Visão Geral dos Algoritmos Online. Adaptado de [36].	78
Figura 5.2 Mudanças na distribuição dos dados em data streaming. Adaptado de [26].	80
Figura 5.3 Pseudocódigo do algoritmo ADWIN.	83
Figura 5.4 Intuição de como funciona o algoritmo AnyNovel.	84
Figura 5.5 Exemplo da formação de clusters pelo algoritmo AnyNovel (lado esquerdo) e a estrutura interna de um cluster e micro cluster (lado direito).	85
Figura 6.1 Visão geral de como ocorre a discretização dos dados e a geração de BOPs.	88

Figura 6.2 Etapas do framework DISTAR. ....	89
Figura 6.3 Visão mais detalhada das etapas do framework DISTAR. ....	89
Figura 6.4 Diagrama BPMN que detalha o passo a passo da algoritmo NOHAR. ....	92
Figura 6.5 Entrada de dados do NOHAR. ....	93
Figura 6.6 (a) representa os dados originais e (b) representa os dados após a aplicação do filtro Exponential Smoothing. OBS: Linha preta é o eixo x, linha azul é o eixo y e linha vermelha é o eixo z. ....	94
Figura 6.7 (a) representa os dados das atividades, sitting e (b) os dados da atividade standing. Esses dados foram extraídos da base de dados UCI-HAR. OBS: Linha preta é o eixo x, linha azul é o eixo y e linha vermelha é o eixo z. ....	95
Figura 6.8 Exemplo do resultado da etapa de discretização. ....	95
Figura 6.9 Exemplo de deslocamento de palavras. ....	96
Figura 6.10 Exemplo do resultado final da etapa de geração de histogramas. ....	97
Figura 6.11 Exemplo de dois histogramas BOSS. ....	97
Figura 6.12 Exemplo de tolerância representada por folgas aceitáveis que podem variar em um histograma de referência. ....	99
Figura 7.1 Tela principal da ferramenta BHAR. Fonte: autor. ....	102
Figura 7.2 Taxa de acurácia do NOHAR ao longo do tempo e dos momentos em que o usuário é solicitado para fornecer o rótulo da atividade. A barra pontilhada que atinge o valor 75 indica uma requisição para a atividade andar e a barra pontilhada que atinge o valor 60 indica as requisições para as demais atividades. ....	114
Figura 7.3 Visão geral das acurácias do NOHAR e AnyNovel ao longo do tempo para as bases de dados SHOAIB, WISDM e UCI-HAR. ....	116
Figura 7.4 Acurácias (linhas superiores dos gráficos) e momentos em que o modelo de classificação solicitou o rótulo das atividades para o usuário (barras pontilhadas) dos algoritmos NOHAR e AnyNovel nas bases de dados SHOAIB, WISDM e UCI-HAR. ....	117
Figura 7.5 Visão geral do tempo de processamento dos algoritmos NOHAR e AnyNovel nas bases de dados SHOAIB, WISDM e UCI-HAR. ....	118
Figura 7.6 Análise comparativas das acurácias ao longo do tempo dos algoritmos NOHAR, HT e HAT para as bases de dados SHOAIB, WISDM e UNI-HAR. ....	120
Figura 7.7 Visão geral do tempo de processamento dos algoritmos NOHAR, Hoeffding Tree e Hoeffding Adaptive Tree nas bases de dados SHOAIB, WISDM e UCI-HAR. ....	121
Figura A-1. Exemplo da distribuição das classes por atividade na base de dados UCI-HAR. ....	145
Figura A-2. Posições do smartphone no corpo do usuário de onde os dados da base Shoaib foram coletados. Fonte [91]. ....	145
Figura A-3. Exemplo da distribuição das classes por atividade na base de dados WISDM. ....	146
Figura C-1. Acurácia dos modelos de classificação baseado nas características de tempo. Fonte: autor. ....	151
Figura C-2. Acurácia dos modelos de classificação baseado nas características de frequência derivadas do FFT. Fonte: autor. ....	152
Figura C-3. Acurácia dos modelos de classificação baseados nas características de frequência derivadas do Wavelet. Fonte: autor. ....	152
Figura C-4. Acurácia dos modelos de classificação baseados nas características de tempo derivadas da magnitude. Fonte: autor. ....	153
Figura C-5. Acurácia dos modelos de classificação baseados nas características de tempo derivadas dos componentes verticais e horizontais. Fonte: autor. ....	153
Figura C-6. Resumo das taxas de acurácias dos modelos de classificação gerados pelo algoritmo Random Forest com e sem a aplicação de filtros. Fonte: autor. ....	155

Figura C-7. Resumo das acurácias dos modelos de classificação após a seleção de características. Fonte: autor.....	156
Figura C-8. Resultados das acurácias dos modelos de classificação com a variação na taxa de sobreposição. Fonte: autor.....	157
Figura C-9. Resultados das acurácias dos modelos de classificação com a variação da janela de tempo. Fonte: autor. ....	157
Figura C-10. Exemplos de acurácia dos modelos de classificação utilizando diferentes bases de dados. Fonte: autor.....	159
Figura C-11. Comparação das acurácias dos modelos de classificação com a combinação dos dados do acelerômetro e giroscópio. Fonte: autor. ....	161
Figura C-12. Comparação entre as taxas de acurácia dos modelos de classificação dos algoritmos de aprendizagem rasa e profunda. Fonte: autor.....	162
Figura D-1. Tela principal da ferramenta BHAR. ....	163
Figura D-2. Tela de Treinamento e classificação. ....	164
Figura D-3. Tela do algoritmo NOHAR. ....	164

# Lista de Tabelas

Tabela 3.1 Lista de trabalhos separados por tipos de coleta de dados. ....	39
Tabela 3.2 Lista de trabalhos separados por frequência. ....	40
Tabela 3.3 Lista de trabalhos separados por posição. ....	40
Tabela 3.4 Lista de trabalhos separados por orientação. ....	41
Tabela 3.5 Lista de bases de dados públicas. A – Acelerômetro, G - Giroscópio e M – Magnetômetro. ....	41
Tabela 3.6 Lista de trabalhos separados por tamanho de janelas. ....	43
Tabela 3.7 Time Features. ....	44
Tabela 3.8 Frequency Features. ....	46
Tabela 3.9 Lista de trabalhos separados por características. ....	48
Tabela 3.10 Lista de Trabalhos que utilizam as características de baixo nível. ....	48
Tabela 3.11 Lista de trabalhos separados por algoritmos de aprendizagem convencionais. ....	51
Tabela 3.12 Lista de trabalhos separados por algoritmos de aprendizagem profunda. ....	52
Tabela 4.1 Exemplo da matriz de pesos gerados pelos TF-IDF. ....	67
Tabela 5.1 Características do modelo do AnyNovel. ....	85
Tabela 7.1 Lista das características do domínio da frequência e do tempo. ....	104
Tabela 7.2 Resultados referentes a acurácia dos modelos de classificação geradas pelos algoritmos rasos combinados com as características do domínio do tempo e da frequência. ...	106
Tabela 7.3 Resultados referentes a acurácia dos modelos de classificação geradas pelos algoritmos de representação simbólica utilizando as estratégias de fusão de coordenadas por empilhamento, magnitude e PCA. ....	106
Tabela 7.4 Lista de tempo de processamento na etapa de extração de características. ....	107
Tabela 7.5 Lista de tempo de processamento na etapa de treinamento do modelo de classificação. ....	108
Tabela 7.6 Consumo de memória em Bytes dos dados antes e depois do processo de extração de características. ....	109
Tabela 7.7 Taxas de acurácia do algoritmo NOHAR para cada usuário da base de dados ShoaiB. ....	112
Tabela 7.8 Matriz de confusão do algoritmo NOHAR. ....	113
Tabela 7.9 Número de histogramas desconhecidos, de treino e teste gerados para a base de dados SHOAI B. ....	113
Tabela 7.10 Taxas de acurácia do algoritmo AnyNovel para cada usuário da base de dados SHOAI B. ....	115
Tabela 7.11 Matriz de confusão do algoritmo AnyNovel. ....	115
Tabela 7.12 Taxas de acurácia do algoritmo Hoeffding Tree o para cada usuário da base de dados SHOAI B. ....	119
Tabela 7.13 Taxas de acurácia do algoritmo Hoeffding Adaptive Tree para cada usuário da base de dados SHOAI B. ....	120
Tabela 7.14 Consumo de memória em Bytes dos dados antes e depois do processo de extração de características do algoritmo NOHAR. ....	122
Tabela C.1 Conjunto de experimentos realizados na primeira etapa. ....	150
Tabela C.2 Matriz de confusão para a estratégia Cross-validation. ....	158



# Lista de Abreviações

ANN	Redes neurais artificiais (Artificial Neural Network)
BOP	Conjunto de padrões simbólicos (Bag-of-Patterns)
BOSS	Bag-Of-SFA-Symbols
BOSS VS	Bag-Of-SFA-Symbols in Vector Space
BoW	bag of words
CNN	Redes neurais de convolução (Convolution Neural Network)
DFN	Deep fully-connected network
DFT	Transformada de Fourier Discreta (Discrete Fourier Transform)
DT	Decision Tree
DNN	Redes neurais profundas (Deep Neural Network)
KNN	Algoritmo K- Vizinhos mais próximos (K-nearest neighbor)
LSTM	Long Short-Term Memory
MCB	Partição de múltiplos coeficientes (Multiple Coeficiente Binning)
MFT	Transformada de Fourier Momentânea (Momentary Fourier Transform)
NB	Algoritmo Naive Bayes
PCA	Análise de componentes principais (Principal Component Analysis)
RAH	Reconhecimento de atividades Humanas
RBM	Restricted boltzmann machine
RNN	Redes neurais recorrentes (Recurrent Neural Network)
SAE	Stacked Autoencoder
SAX	Aproximação simbólica agregada (Symbolic Aggregate Approximation)
SAX-VSM	Symbolic Aggregate Approximation Vector Space Model
SFA	Aproximação simbólica de Fourier (Symbolic Fourier Approximation)
SVM	Algoritmo Máquina de vetores de suporte (Suporte Vector Machine)
tf-idf	Frequência de termos – frequência de termos invertidos
UCI-HAR	Machine Learning Repository: Data Sets – Human Activity Recognition
WISDM	WISDM: WIreless Sensor Data Mining
WEASEL	Word extraction for time series classification

# CAPÍTULO 1

## INTRODUÇÃO

As atividades humanas têm sido comumente utilizadas na definição do padrão de comportamento humano. A disponibilidade dos sensores em plataformas móveis tem possibilitado o desenvolvimento de diversas aplicações práticas em diversas áreas do conhecimento [1] [2] como, por exemplo:

- Saúde – por meio de sistemas de detecção de quedas [3], monitoramento de pessoas idosas [4] e prevenção de doenças [5].
- Internet das Coisas e Cidades Inteligentes – por meio de soluções usadas para reconhecer e monitorar atividades domésticas [6], economia de energia elétrica [7], modelagem de ambientes domésticos [8] e medição do nível de poluição de um ambiente [9].
- Segurança – por meio de soluções de monitoramento de atividades de um indivíduo [10], detecção de anomalias em multidões [11] e rastreamento de objetos [12].
- Transporte – por meio de soluções relacionadas a navegação de veículos [13] [14] e pedestres [15].

Por este motivo, o desenvolvimento de soluções para reconhecer atividades humanas (HAR) por meio de tecnologias e métodos computacionais tem sido bastante explorado nos últimos anos [13] [16] [17] [18] [19]. Nesse sentido, o problema de HAR tem sido tratado pela literatura como um típico problema de reconhecimento de padrão, mais especificamente, voltado para o problema de classificação, ou seja, identificar qual a atividade que está sendo executada por um indivíduo em um determinado momento. Neste caso, as atividades humanas são definidas como um conjunto de ações que podem ser repetidas ao longo do tempo [20]. Quando essas ações se tornam perceptíveis e frequentes, esse conjunto pode ser considerado uma atividade como, por exemplo, andar, dirigir ou cozinhar. Por esta razão, a maioria das soluções de HAR tem sido desenvolvida com auxílio de métodos e técnicas de inteligência artificial, incluindo os algoritmos de aprendizagem de máquina *offline* e *online*. Os algoritmos de aprendizagem *offline* (ou *batch*) examinam um grande volume de dados de uma única vez e induzem um único modelo de predição. Neste caso, a incorporação de novos dados requer a reconstrução do modelo de classificação a partir do processamento de toda a base de dados, tornando esse processo lento. Os algoritmos de aprendizagem *offline* são representados pelos algoritmos rasos (e.g. SVM, Árvore de Decisão, *Naive Bayes* e KNN) [21] [22] [23] e pelos algoritmos profundos (e.g. CNN, RNN, RBM, SAE, DFN e DBM) [24]. Por outro lado, os algoritmos de aprendizado *online* são representados pelos algoritmos de análise de fluxo contínuo de dados (do inglês, *data streaming*), onde os dados são processados à medida que os mesmos chegam no sistema [25] [26].

A qualidade das soluções de reconhecimento de atividades humanas está relacionada com a capacidade preditiva dos modelos de classificação adotados. Tais modelos devem ser capazes de lidar com as particularidades de cada indivíduo, visto que as atividades são executadas de diferentes formas devido às influências relacionadas, principalmente, com hábitos, preferências pessoais e saúde. Além disso, a quantidade de atividades executadas por um humano é muito maior do que as soluções de HAR atuais são capazes de reconhecer.

Mesmo com essas limitações, as soluções desenvolvidas nessa área têm apresentado resultados interessantes em aplicações específicas relacionadas ao bem-estar humano por meio do reconhecimento de atividades físicas dos usuários [24] [27] [28] através dos smartphones.

Os smartphones têm sido comumente empregados no desenvolvimento de soluções de HAR devido à capacidade ubíqua e à diversidade de sensores embutidos nesses dispositivos. Essas características tornaram os smartphones uma plataforma fundamental para a área de HAR, principalmente por serem utilizados por bilhões de pessoas ao redor do mundo [18]. Lane et al. [17] citam quatro fatores que demonstram que o smartphone é uma plataforma ideal para reconhecer atividades humanas. Primeiro, o smartphone é um dispositivo barato que reúne diversos sensores de hardware (e.g. acelerômetro, *bluetooth* GPS) e software (e.g. aplicativos) em um mesmo dispositivo. Segundo, os smartphones são dispositivos abertos e programáveis. Terceiro, os smartphones possuem um alto poder de alcance em massa por meio da distribuição de conteúdo e aplicações via lojas virtuais (e.g. *app store*). Por último, a computação em nuvem permite que desenvolvedores tenham recursos extras que podem servir como suporte para compartilhamento de informações. Além disso, os dados sobre atividades, preferências e interações sociais dos usuários podem ser utilizados para reconhecer, monitorar e reagir a praticamente qualquer fenômeno, seja físico, psicológico ou social.

Em resumo, a vantagem dos smartphones está associada à sua capacidade de a) capturar e processar os dados de sensores; b) transmitir e receber dados via rede; e c) conectar com outros dispositivos disponíveis no ambiente físico. Os sensores inerciais embutidos nos smartphones (e.g. acelerômetro e giroscópio), por exemplo, são utilizados para capturar informações relacionadas a aceleração e a direção do movimento do corpo humano. Esses sensores têm permitido a extração de diversas informações em torno do usuário que podem ser utilizadas para reconhecer atividades físicas individuais [29].

Dentre os diversos sensores disponíveis nos smartphones, as soluções de HAR que utilizam os sensores inerciais para reconhecer as atividades físicas dos usuários têm crescido bastante nos últimos anos. Esse crescimento tem sido evidenciado pela disponibilização de diversas coleções de dados públicas [24] com o propósito de auxiliar na validação das soluções que surgem ao longo do tempo. Por esse motivo, esta tese de doutorado foca somente na análise de dados dos sensores inerciais embutidos nos smartphones e, conseqüentemente, no reconhecimento das atividades físicas e individuais dos usuários como, por exemplo, andar, sentar, ficar em pé, deitar, subir e descer escadas.

## 1.1. Problema

Esta seção apresenta os problemas que evidenciam os principais desafios que ainda precisam ser superados na área de HAR baseada em smartphones. Em resumo, os desafios abrangem dois pontos principais. O primeiro trata da flexibilidade e a escalabilidade das soluções em se adaptarem às mudanças de dados, principalmente quando surgem novos usuários, atividades e ambientes. O segundo trata do alto custo computacional relacionado ao consumo de memória, processamento e energia das baterias.

Em relação ao primeiro problema, a maioria dos trabalhos na literatura utiliza algoritmos de classificação *offline* rasos [22] e profundos [24] para desenvolver soluções de HAR para smartphones. As desvantagens da abordagem *offline* é que os modelos de classificação são limitados à distribuição do histórico de dados e a etapa de geração desses

modelos possui alto custo computacional. Em resumo, esses algoritmos examinam um grande volume de dados uma única vez e induzem um único modelo de predição (ou classificação), o qual não é mais revisado por novos dados. Quando novas informações não previstas no histórico de dados surgem, não é trivial incorporar os novos dados no sistema de aprendizado *offline*, pois isso requer a reconstrução do modelo de previsão a partir do processamento de toda a base de dados. Isso mostra que essa área apresenta restrições em que o aprendizado *offline* não é adequado para compor as soluções, visto que os modelos sofrem degradações ao longo do tempo devido os dispositivos estarem imersos em ambientes físicos amplos (e.g. casa, trabalho, rua, shopping) e completamente dinâmicos.

Além disso, o problema relacionado aos históricos de dados limitados, no contexto dos smartphones, impacta diretamente na geração de modelos de classificação com baixa representatividade da diversidade populacional de usuários [25] [26]. Para resolver esse problema, é necessário que especialistas realizem um enorme trabalho manual de rotulação de dados pertencentes aos representantes dos diversos perfis de usuários existentes na população. No entanto, essa é uma tarefa árdua e, mesmo que fosse realizada, os modelos de classificação perderiam precisão rapidamente ao longo do tempo devido às constantes mudanças nos dados causadas pelas alterações de ambiente e comportamento dos usuários. Nessa perspectiva, as soluções de HAR poderiam se tornar obsoletas rapidamente e, além do mais, seriam limitadas em termos de escalabilidade.

O segundo problema está relacionado com a limitação de recursos computacionais dos smartphones (memória, processamento e energia). A maioria das soluções existentes requer um alto custo no consumo desses recursos, principalmente quando se trata de tarefas como a geração de modelos de classificação e preparação de dados. Por essa razão, as soluções existentes baseadas em algoritmos de aprendizado *offline* tendem a depender de serviços externos para armazenar o histórico de dados e gerar os modelos de classificação [30] [31] [32]. No entanto, essa prática também possui desvantagens nos smartphones, visto que o uso de recursos de rede (e.g. WiFi) requer um consumo maior de energia dos dispositivos para enviar e receber dados. Todos esses problemas evidenciam a dificuldade dos algoritmos *offline* no contexto de HAR para smartphones.

## 1.2. Hipótese

Os problemas apresentados na seção anterior tem motivado o início da concepção de sistemas HAR baseados em aprendizagem *online* [33] [25] [26] [34]. De modo geral, esses sistemas são compostos por um conjunto de algoritmos projetados para analisar fluxo contínuo de dados (do inglês, *data streaming*) em diferentes velocidades, escalas e granularidades [35]. A estratégia utilizada por esses algoritmos permite que os modelos de classificação sejam atualizados dinamicamente à medida que novos dados são coletados. Os algoritmos *online* não dependem de um histórico de dados para gerar os modelos porque somente as informações fundamentais dos dados (e.g. medidas estatísticas) são extraídas e armazenadas na memória. Dessa forma, os algoritmos *online* são capazes de atuar com uma pequena quantidade de memória. Sempre que os dados brutos contidos na memória são processados, a memória é liberada para armazenar os novos dados coletados a partir do fluxo contínuo de dados [36]. O baixo consumo de memória viabiliza a disseminação de aplicações móveis mais consistentes para os smartphones. No contexto dos smartphones, os modelos adaptativos são especializados no usuário, possibilitando que as soluções sejam capazes de aprender as particularidades de cada usuário de modo que

decisões mais específicas possam ser tomadas. Isso resolve o problema relacionado à diversidade de hábitos, comportamentos e saúde dos usuários, além de possibilitar a escalabilidade da solução.

Além da escolha dos algoritmos de classificação, é necessário observar também as técnicas de preparação de dados mais adequadas para atuar no contexto dos smartphones. Nesse cenário, o desafio é encontrar um compromisso entre eficiência e o consumo de recursos computacionais. Atualmente, as soluções de HAR têm seguido uma metodologia padrão com etapas bem definidas para o processamento dos dados, como coleta, segmentação e fusão de dados, extração e seleção de características e treinamento dos modelos de classificação. Em especial, a etapa de extração de características é a mais explorada pela literatura de HAR porque, de fato, a escolha dos atributos (ou características) impacta diretamente na qualidade dos modelos de classificação. A literatura de HAR divide as características extraídas de sensores inerciais em três domínios diferentes [37]: tempo, frequência e discreto. As características do domínio do tempo e frequência são amplamente utilizadas pelos algoritmos de aprendizagem *offline* rasos. Tais características são extraídas, por exemplo, através de cálculos matemáticos e estatísticos aplicados em cada segmento da série temporal (ou sinais dos sensores inerciais). Por outro lado, as características do domínio discreto são extraídas por meio da discretização de dados utilizando algoritmos de representação simbólica, como os algoritmos SAX (*Symbolic Aggregate Aproximation*) [38] e o SFA (*Symbolic Fourier Aproximation*) [39]. No contexto desta tese, a discretização de dados consiste na transformação de segmentos de sinais dos sensores inerciais em símbolos (ou palavras), como descrito no Capítulo 4.

Em especial, as características do domínio discreto possuem três vantagens em relação as características dos outros domínios. A primeira vantagem consiste na natureza automática de redução de ruídos e dimensionalidade dos dados. A segunda vantagem consiste na redução do consumo de recursos computacionais como memória e processamento. Por fim, a terceira vantagem consiste na capacidade de gerar bons modelos de classificação na perspectiva de séries temporais [40] [41] [42] [43]. Baseado nisso, as características do domínio discreto evidenciam dois benefícios que simplificam a metodologia padrão utilizada no processamento de dados dos sensores inerciais. O primeiro benefício consiste na eliminação de um processamento extra na etapa de segmentação de dados, onde filtros (e.g. passa-baixa ou média móvel) são utilizados para reduzir o ruído do sinal. O segundo benefício consiste na eliminação de outro processamento extra na etapa de seleção de características, onde técnicas de redução da dimensionalidade dos dados (e.g. ganho de informação ou PCA) são utilizadas para selecionar as melhores características da base de dados. Esses benefícios são evidentes porque os algoritmos de representação simbólica são, naturalmente, redutores de ruídos e dimensionalidade de dados ao mesmo tempo.

Tendo em vista as vantagens dos algoritmos de aprendizagem *online* e dos algoritmos de representação simbólica, abordados acima, é evidente que ambas estratégias são mais apropriadas para desenvolver soluções de HAR no contexto dos smartphones equipados com sensores inerciais. Efetivamente, existem poucos trabalhos recentes na literatura que tratam especificamente da análise do fluxo contínuo de dados de sensores inerciais embutidos nos smartphones [33] [25] [26] [34] e, também, poucos trabalhos que utilizam algoritmos de representação simbólica na preparação desses dados [44] [45]. Por outro lado, nenhum trabalho na literatura utiliza ambas as estratégias conjuntamente. Isso evidencia uma boa oportunidade de exploração científica e avanço do estado da arte nesse âmbito. Nesse contexto, essa tese de doutorado se baseia na seguinte hipótese:

*A combinação entre algoritmos de aprendizagem online e algoritmos de representação simbólica pode gerar uma nova solução adaptativa, de baixo custo e escalável, capaz de manipular uma quantidade massiva de dados com poucos recursos computacionais de memória e processamento.*

Além das evidências discutidas acima que reforçam a veracidade dessa hipótese, a literatura, de modo geral, mostra que ainda não existe um algoritmo *online* capaz de aprender dados simbólicos [35]. Nesse sentido, Gama et al. [35] recomendam utilizar algoritmos de discretização no contexto da análise de fluxo contínuo de dados para aplicações em tempo real. O motivo dessa recomendação é exatamente o desenvolvimento de soluções de baixo custo computacional, tanto que em seu trabalho os algoritmos recomendados são o PiD [46] e o SAX [38] descritos na Seção 3.5.2. Baseado nisso, esta tese de doutorado explora um novo caminho baseado na aprendizagem de dados simbólicos no contexto da análise de fluxo contínuo de dados. Esse caminho dá origem a uma nova metodologia para o desenvolvimento de algoritmos *online* de baixo custo que manipulam dados simbólicos, além de criar uma nova categoria de algoritmos de aprendizado simbólico *online*.

Os resultados desta tese de doutorado contribuem para o avanço do estado da arte em duas áreas distintas. A primeira é na área de HAR com o desenvolvimento de uma solução mais aprimorada. A segunda é na área de análise de fluxo contínuo de dados (*Data Streaming*) com um algoritmo simbólico *online* pioneiro.

Nesse contexto, esta tese de doutorado combina as vantagens dos algoritmos de aprendizagem *online* e dos algoritmos de representação simbólica em uma única solução, além de demonstrar o uso isolado dessas estratégias para HAR. Por exemplo, no caso dos algoritmos *online*, será mostrado o comportamento dos algoritmos existentes na literatura como o Hoeffding Tree [47]. No caso dos algoritmos de representação simbólica, será mostrado o comportamento dos algoritmos de classificação simbólicos SAX-VSM [40], BOSS-VS [41] e WEASEL [43]. O objetivo é facilitar o entendimento das vantagens de cada uma dessas estratégias em relação às abordagens tradicionais baseadas nas características do domínio do tempo e da frequência.

### 1.3. Objetivo

Neste contexto, este trabalho tem como objetivo aprimorar as soluções de HAR baseadas em smartphones instrumentados com sensores inerciais por meio do uso dos algoritmos *online* e dos algoritmos de representação simbólica dos dados. A meta é desenvolver uma solução adaptativa, escalável e de baixo custo com o propósito de resolver os problemas apontados na Seção 1.1. Para isso, esta tese de doutorado propõe uma abordagem pioneira que combina as vantagens da aprendizagem *online* com as vantagens dos algoritmos de representação simbólica. A abordagem proposta é composta pelo *framework* DISTAR (*Discrete Stream learning for Activity Recognition*) e pelo algoritmo *online* adaptativo NOHAR (*NOvelty discrete data stream for Human Activity Recognition*). O DISTAR é um *framework* adaptado para suportar algoritmos de representação simbólica na perspectiva da abordagem de *data streaming* e o NOHAR é um algoritmo simbólico *online*.

Para alcançar esse objetivo, os seguintes objetivos específicos devem ser atingidos:

1. Projetar e desenvolver um *framework* para fornecer uma metodologia para o desenvolvimento de futuros algoritmos simbólicos *online* com o propósito de padronizar as etapas de processamento de dados.
2. Projetar e desenvolver um algoritmo *online* simbólico de baixo custo como prova de conceito do *framework* proposto.

## 1.4. Contribuições

A primeira contribuição desta tese de doutorado consiste no projeto e desenvolvimento do *framework* DISTAR e do algoritmo simbólico *online* NOHAR. Ambas as propostas são baseadas em conceitos de aprendizagem *online* e técnicas de representação simbólica. O objetivo é combinar as vantagens de ambas as estratégias e desenvolver um método escalável e de baixo custo capaz de manipular uma quantidade massiva de dados. Além disso, essa solução deve ser capaz de atuar em um ambiente limitado por memória e processamento. A inovação por trás dessa solução atinge dois pontos. O primeiro é o desenvolvimento de um algoritmo *online* pioneiro capaz de manipular dados simbólicos e o segundo é a criação de uma nova categoria de algoritmos *online* para a área de *data streaming*.

A segunda contribuição está relacionada com uma nova proposta para uso de algoritmos de representação simbólica na área de HAR. Essa proposta está fundamentada na capacidade desses algoritmos reduzir a dimensionalidade dos dados, permitindo a manipulação de uma quantidade massiva de dados [48]. Nesse sentido, esta tese utiliza dois algoritmos de representação simbólica, *Symbolic Aggregate Approximation* (SAX) [38] e *Symbolic Fourier Approximation* (SFA) [39]. Ambos os algoritmos são bases para os algoritmos de classificação SAX-VSM [40], BOSS-VS [41] e WEASEL [43]. Além disso, esta tese propõe duas estratégias para o tratamento de séries temporais multidimensionais devido à natureza dos dados dos sensores inerciais. O propósito dessas estratégias é evitar alterações desnecessárias nos algoritmos originais focados somente no tratamento de séries temporais unidimensionais (ver Seção 4.6). As análises dos algoritmos de representação na área de HAR foram publicadas na revista *Sensors*. No primeiro trabalho [49], nós apresentamos uma análise comparativa dos algoritmos de classificação simbólicos com os principais trabalhos da área de HAR e o segundo trabalho [50] descreve uma adaptação do algoritmo BOSS [41], denominado de MBOSS (*Multivariate Bag-of-Symbols*), para manipular múltiplas séries temporais no contexto de HAR. O MBOSS foi desenvolvido em uma dissertação vinculada a esta tese de doutorado. Além disso, outra avaliação do método MBOSS também foi publicada em [51]. Em paralelo, o estudo dos algoritmos de representação simbólica gerou um capítulo de livro focado em detecção de anomalias em séries temporais [52].

A terceira contribuição desta tese trata de uma apresentação mais aprimorada do estado da arte de HAR no contexto dos smartphones com sensores inerciais. Essa apresentação compreende a descrição: 1) do entendimento das atividades físicas dos usuários; 2) do entendimento dos dados dos sensores inerciais; 3) da preparação dos dados; 4) do processo de modelagem e inferência dos dados; e 5) da avaliação das soluções de HAR. Essas etapas são parte de uma metodologia bem definida, comumente utilizada no desenvolvimento das soluções de HAR. Para cada etapa da metodologia, é apresentada uma descrição detalhada do processo juntamente com uma lista de trabalhos agrupados pelas diferentes formas de tratamentos dos dados e usos de técnicas específicas aplicadas em cada etapa. Além disso, são destacadas as recomendações das melhores práticas de implementação das técnicas para cada etapa.

A quarta contribuição consiste no desenvolvimento de uma ferramenta que centraliza a implementação dos *baselines* mapeados na literatura, inclusive o método proposto por esta tese, com o propósito de facilitar os experimentos, validações e comparações entre os métodos.

## 1.5. Organização do Documento

Os capítulos restantes desta tese estão organizados de forma que o leitor tenha um entendimento detalhado da área de HAR antes de apresentarmos a proposta desta tese. Para isso, a fundamentação teórica foi dividida em dois capítulos. O primeiro descreve os conceitos e o segundo a metodologia padrão de HAR. O próximo capítulo aborda detalhes das técnicas de representação simbólica dos dados e os restantes dos capítulos abordam sobre o aprendizado *online*, proposta e experimentos e avaliações. A seguir são apresentados um resumo do que será tratado em cada capítulo.

2º Capítulo – o segundo capítulo trata dos principais conceitos necessários para o entendimento das atividades humanas baseado em smartphones com sensores inerciais. Além disso, o capítulo mostra a evolução da área de HAR no contexto dos smartphones com o propósito destacar os principais trabalhos que marcaram eventos importantes ao longo dos anos. Em seguida, o capítulo apresenta uma visão geral das abordagens comumente empregada nas soluções de HAR. Depois disso, os principais desafios, oportunidades e soluções viáveis são abordados com base nas dificuldades e lacunas mapeadas na literatura. Por fim, são apresentadas as principais aplicações práticas na qual as soluções podem ser empregadas.

3º Capítulo – o terceiro capítulo apresenta detalhes das etapas da metodologia padrão comumente empregada nas soluções de HAR. Na etapa de coleta de dados, o capítulo aborda como os dados dos sensores inerciais são tratados em termos de frequência da coleta, posição e orientação do smartphone no corpo do usuário. Na etapa de preparação dos dados, o capítulo descreve um conjunto de transformações nos dados necessárias para atuação dos algoritmos de modelagem e inferência de dados. Nesse sentido, o capítulo aborda as principais técnicas de segmentação de dados e o processo de extração de características. Na etapa de modelagem, o capítulo apresenta os algoritmos de classificação rasos e profundos comumente utilizados para gerar modelos de classificação das atividades humanas. Por fim, é apresentado as principais técnicas de fusão de dados utilizadas na área de HAR.

4º Capítulo – o quarto capítulo apresenta conceitos relacionados aos algoritmos de representação simbólica responsáveis por extrair características no domínio discreto. Em seguida, são apresentados os algoritmos de classificação simbólicos focados em manipular dados discretos. Tais algoritmos incluem SAX-VSM, BOSS-Model, BOSS-VS e WEASEL. Por fim, este capítulo apresenta várias estratégias de manipulação de séries temporais multidimensionais que podem ser utilizadas para tratar os dados de sensores inerciais, inclusive novas propostas adicionais provenientes desta tese de doutorado.

5º Capítulo – o quinto capítulo apresenta os fundamentos relacionados a aprendizagem *online*. O capítulo inicia com uma visão geral das principais diferenças entre a aprendizagem *offline* e a *online*, incluindo a apresentação das vantagens do emprego dessa abordagem no contexto de HAR para smartphones. Em seguida, o capítulo mostra uma visão geral dos algoritmos *online* por meio da apresentação de conceitos. Depois, uma nova tendência desses algoritmos baseada na detecção de novidades ao longo do tempo é apresentada. Após isso, são apresentados os trabalhos relacionados na perspectiva dos smartphones com sensores inerciais.



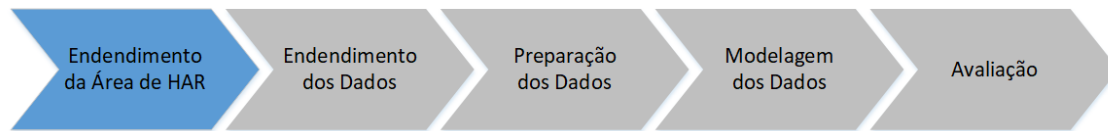
Neste ponto, são discutidas as estratégias utilizadas e as vantagens e desvantagens de cada uma delas. Por fim, o capítulo aborda as estratégias de avaliação dos algoritmos *online*.

6º Capítulo – o sexto capítulo apresenta a proposta principal desta tese de doutorado composta pelo o *framework* DISTAR e pelo algoritmo *online* simbólico NOHAR. Nossa proposta é baseada em uma nova estratégia relacionada a análise de fluxo contínuo de dados discretos. Inicialmente, são apresentados os motivos pelos quais este caminho foi escolhido. Em seguida, são apresentados os detalhes referentes ao *framework* DISTAR e ao algoritmo NOHAR.

7º Capítulo – o sétimo capítulo apresenta um conjunto de avaliações dos métodos e técnicas discutidos neste trabalho. Tais avaliações são divididas em três partes. A primeira parte trata da avaliação dos algoritmos rasos e profundos com enfoque nas etapas da metodologia de HAR. A segunda parte trata da avaliação dos algoritmos *offline* de representação simbólica em comparação com os algoritmos *offline* rasos. Por fim, a terceira parte trata da avaliação do algoritmo NOHAR em comparação dos os algoritmos AnyNovel, *Hoeffding Tree* e *Hoeffding Adaptive Tree*.

# CAPÍTULO 2

## RECONHECIMENTO DE ATIVIDADES HUMANAS



Este capítulo trata dos principais conceitos necessários para o entendimento das atividades humanas baseado em smartphones com sensores inerciais. Além disso, o capítulo mostra a evolução da área de HAR no contexto dos smartphones com o propósito destacar os principais trabalhos que marcaram eventos importantes ao longo dos anos. Em seguida, o capítulo apresenta uma visão geral das abordagens comumente empregada nas soluções de HAR. Depois disso, os principais desafios, oportunidades e soluções viáveis são abordados com base nas dificuldades e lacunas mapeadas na literatura. Por fim, são apresentadas as principais aplicações práticas na qual as soluções podem ser empregadas.

### 2.1. Entendendo as Atividades Humanas

Atividades podem ser definidas como sendo um conjunto de ações executadas pelo usuário durante um determinado período de tempo como tomar banho, cozinhar, trabalhar e dirigir [20]. Formalmente, uma instância ou ocorrência de uma atividade é definida por uma sequência de eventos. Os eventos são definidos como sendo sequências de dados formados por consecutivas leituras de sensores em um determinado período de tempo. Cada evento,  $e$ , possui a forma  $e = (t, s, m)$  onde  $t$  representa o tempo,  $s$  o sensor e  $m$  a mensagem do sensor. Dessa forma, um conjunto de atividades pode ser definido como  $A = (a_1, a_2, \dots, a_n)$ , onde  $a_n = (e_1, e_2, \dots, e_k)$  representa a  $n$ -ésima atividade e  $e_k$  o  $k$ -ésimo evento [24].

As atividades humanas podem ser categorizadas pelo nível de complexidade do processo de reconhecimento. Dernbach et al. [53] e Shoaib et al. [27], por exemplo, categorizam as atividades como simples e complexas, enquanto Khan et al. [54] e Reiss [55] categorizam as atividades como sendo de baixo e alto nível. Ambas as categorias possuem o mesmo significado. De modo resumido, as atividades simples ou de baixo nível são aquelas atividades que podem ser reconhecidas apenas com a análise de dados de um ou mais sensores em um curto período de tempo (e.g. andar e correr). Enquanto as atividades complexas ou de alto nível podem ser vistas como um conjunto de atividades de baixo nível que podem ser reconhecidas por um longo período de tempo (e.g. trabalhar e fazer compras).

A maioria dos estudos de RAH em smartphones concentra-se, sobretudo, no uso dos sensores inerciais, acústicos, rádio e de localização. Baseado nisso, as atividades nesta escala podem ser divididas em dois grupos [13] [22]. O primeiro trata das atividades relacionadas ao movimento do indivíduo (e.g. andar e correr) e o segundo trata das atividades relacionadas a localização do indivíduo (e.g. trabalhar e fazer compras). Estudos relacionados as atividades de movimento focam na análise das atividades físicas dos usuários [29], enquanto que os estudos

relacionados as atividades de localização focam no rastreamento da posição dos usuários [56] [57] [15].

## 2.2. Entendendo a Evolução da Área de HAR

A seguir são apresentados o contexto histórico e os principais eventos históricos que marcaram as evoluções da área de HAR na perspectiva dos smartphones. Esta seção tem como propósito registrar e reforçar a motivação do uso desses dispositivos na área de HAR.

### 2.2.1 Contexto Histórico

O contexto histórico mostra que a área de HAR surgiu com o advento da computação ubíqua e pervasiva com Mark Weiser em 1991 [58]. A partir desse período, a área de HAR tem evoluído gradualmente conforme surgem novas tecnologias de hardware, para sensoriamento de ambientes, e técnicas computacionais, para manipulação dos dados. Durante três décadas, a comunidade científica tem unido forças para traduzir as atividades dos usuários executadas no mundo físico para o mundo digital por meio da combinação de dados de sensores e técnicas computacionais [20]. Nesse contexto, a área de HAR tem atuado em duas perspectivas: 1) ambientes inteligentes instrumentados com câmeras de vídeo, sensores e atuadores; e 2) usuários equipados com sensores vestíveis (*body-worn sensors*) incluindo usuários que portam smartphones.

O reconhecimento de atividades baseadas em ambientes inteligentes [59] consiste no monitoramento de variáveis do ambiente físico (e.g. pressão, temperatura e movimento) por meio da análise de dados de sensores digitais, como câmeras de vídeo. As interações dos usuários nesses ambientes geram dados capazes de discriminar as ações dos usuários de forma que seja possível identificar as atividades que estão sendo executadas. Dessa forma, as informações obtidas são utilizadas para auxiliar as tomadas de decisões em aplicações específicas como, por exemplo, economizar energia elétrica do ambiente com base nas atividades dos usuários [60] ou monitorar comportamentos humanos suspeitos no ambiente [10]. Por outro lado, os sensores vestíveis (e.g. *smartwatches* e *smart glasses*) permitem o reconhecimento de atividades humanas de forma mais granular, ou seja, é possível reconhecer atividades no âmbito dos movimentos do corpo humano (e.g. andar e pular). Além disso, os sensores vestíveis permitem maior flexibilidade quanto a atuação no ambiente, pois as atividades podem ser reconhecidas em qualquer lugar, independentemente se o ambiente tenha sido instrumentado ou não. Dessa forma, outras novas oportunidades de aplicações surgiram com os sensores vestíveis como monitoramento das atividades físicas e cuidados com a saúde dos usuários [61] [62].

Na última década, muitas soluções para HAR têm sido baseadas em smartphones. Tal fato tem reforçado e estimulado o desenvolvimento de novas pesquisas na área de HAR nos seguintes cenários: a) reconhecimento de atividades físicas dos usuários [63] [64]; b) monitoramento da rotina dos usuários [65] [66]; e c) reconhecimento de atividades em grupos ou comunidades [67] [68]. Além disso, as atividades de grupos têm impulsionado pesquisas na área de sensoriamento de multidões [69] que, por sua vez, tem contribuído para pesquisas relacionadas à Internet das Coisas (IoT) [70] e cidades inteligentes [71]. Portanto, os smartphones têm sido considerados a plataforma móvel mais completa para reconhecer atividades humanas em todas as perspectivas, incluindo descoberta, reconhecimento e previsão das atividades dos usuários.

### 2.2.2 Eventos Históricos baseados nos Smartphones

O primeiro marco histórico foi em 2006 quando surgiram as primeiras soluções de HAR utilizando explicitamente smartphones [72] [73]. Nessa época, os primeiros estudos foram realizados com análise de dados extraídos dos sensores GSM e acelerômetro com o propósito de monitorar a mobilidade dos usuários. Nesse período, todo o processamento de dados era realizado em um computador (processamento *offline*) devido os dispositivos móveis possuírem poucos recursos computacionais.

A partir de 2007, a literatura avançou para o desenvolvimento das primeiras soluções colaborativas [30] [31] [32]. O modelo de comunicação e processamento adotados por essas soluções segue o seguinte fluxo: os dados dos smartphones eram coletados, enviados para um servidor na Internet, onde as informações dos usuários eram compartilhadas. Em geral, tais informações eram utilizadas para melhorar a precisão dos modelos de classificação dos algoritmos de aprendizagem de máquina. Somente a partir de 2008, com a evolução das tecnologias de processamento e armazenamento dos smartphones, começaram a surgir soluções em que a coleta e processamento dos dados são executados no próprio smartphone [74].

A partir de 2009, os smartphones passaram a ser equipados com um número ainda maior de sensores, permitindo o reconhecimento de novas atividades dos usuários como reconhecimento de voz humana. Os trabalhos SoundSense [75] e CenceMe [67] representam bem esse cenário com reconhecimento de atividades baseadas no som ambiente como conversas de grupo. Além disso, neste ano surgiram trabalhos mais voltados para o desenvolvimento de aplicações na área da saúde como detecção de doenças crônicas baseadas nos problemas de locomoção do usuário [76] [77] [4].

A partir de 2010, o enfoque das pesquisas foi dado para o melhoramento de especificidades do processo de reconhecimento de HAR como: i) coleta de dados eficiente por meio do sensoriamento contínuo, com o propósito de melhorar a eficiência energética das baterias [78]; ii) Aprimoramento dos modelos de classificação usando um conjunto de classificadores [79]; iii) detecção dos intervalos de transição entre atividades [64]. Além disso, Berchtold et al. [31] propuseram a criação de um serviço de HAR na nuvem que permite uma constante atualização dos modelos de classificação embarcados nos smartphones por meio do *feedback* dos usuários.

A partir de 2011, surgiram soluções mais específicas como, por exemplo, as primeiras discussões relacionadas com os efeitos da posição e orientação dos smartphones nas soluções de HAR [29] [80]. Por exemplo, Henpraserttae et al. [80] mostram que a localização e orientação do smartphone na mão e na cintura do indivíduo produz sinais diferentes e, conseqüentemente, requerem diferentes análises para reconhecer uma mesma atividade. Além disso, nesse ano ocorreram as primeiras publicações de base de dados públicas como a base de dados WISDM [81]. As bases de dados públicas auxiliam na validação e na comparação de novos métodos de HAR em relação aos existentes.

Em 2012 surgiram os primeiros trabalhos voltados para o reconhecimento de atividades mais complexas utilizando smartphones. Dernbach et al. [53] e Khan et al. [54], por exemplo, combinaram os dados dos sensores inerciais para reconhecer atividades domésticas (e.g. cozinhar) e físicas, respectivamente. Das et al. [82] combinam dados dos sensores dos smartphones e dos sensores de ambientes inteligentes para reconhecer atividades diárias dos usuários. Outros trabalhos focam no aperfeiçoamento de soluções online de forma que todo o processo dos dados ocorra no dispositivo móvel [83] [84].

A partir de 2013, técnicas de fusão de dados para múltiplos sensores começaram a ser aplicadas na etapa de extração de características [85]. Além disso, estudos mais detalhados com o propósito de descobrir características mais acuradas e de baixo custo foram realizados [86]. Em 2014, algumas pesquisas focaram no uso das técnicas de segmentação dos dados (ver Seção 3.1) com objetivo de mensurar o impacto delas na precisão dos modelos de classificação [87]. A partir de 2015, a área de HAR começou a convergir para a aplicação dos métodos de aprendizagem profunda com o primeiro trabalho desenvolvido por Alsheikh et al. [88].

Em 2016, surgiram novas pesquisas de HAR voltadas para análise de fluxo contínuo de dados (*data streaming*) [89]. Nesse ponto, o problema é tratado na perspectiva da aprendizagem *online*. Algumas das vantagens dessa abordagem incluem a não necessidade de armazenamento de um histórico de dados para treinar os modelos de classificação um uso menor de recursos como memória e processamento do smartphone. Além disso, esse método se baseia na detecção de novidades com o objetivo de mapear novas atividades que, porventura, não estejam previstas nos modelos de HAR. O método inclui também uma interação contínua com o usuário através da aprendizagem ativa. Também, no mesmo ano surgiram novos estudos [90] relacionados ao reconhecimento de transição entre atividades (e.g. *sit-to-stand*) na tentativa de eliminar erros de classificação dos modelos, pois os dados referente as transições são considerados ruídos na base de dados.

Em 2017, surgiram diversas pesquisas [91] [92] [93] [54] [94] que realizaram análises comparativas entre as variadas soluções de HAR disponíveis na literatura. A maioria das análises busca descobrir a efetividade das características nos modelos de classificação gerados pelos algoritmos de aprendizagem de máquina. Recentemente, em 2018, começaram a surgir *frameworks* com intuito de encapsular todas as etapas e métodos estudados ao longo do tempo na literatura em uma arquitetura genérica [95]. A partir daí, novas implementações de HAR em formato de API (*Application Programming Interface*) poderão surgir para facilitar o desenvolvimento das aplicações de HAR.

### 2.3. Sensores nos Smartphones

Os sensores são dispositivos eletrônicos que traduzem algumas informações do ambiente real para o ambiente virtual. Tais informações incluem, por exemplo, a temperatura e a umidade do ambiente. Os smartphones possuem diversos sensores embutidos os quais podem ser classificados da seguinte forma [23]: sensores inerciais (e.g. acelerômetros e giroscópios), magnéticos (e.g. magnetômetro), acústicos (e.g. microfone), óticos (e.g. infravermelhos), atmosféricos (e.g. temperatura, barômetro, umidade, gravidade e pressão), ambiente (e.g. luz e proximidade), localização (e.g. GPS), vídeo e imagem (e.g. câmeras), rádio (e.g. GSM, WiFi e Bluetooth) e fisiológicos (e.g. sensores de frequência cardíaca).

Esses sensores possuem diferentes arquiteturas e padrões de dados e são utilizados para diferentes propósitos. Por exemplo, os sensores inerciais são utilizados para capturar os movimentos dos usuários. Os sensores acústicos medem a vibração e sinais sonoros permitindo o reconhecimento de vozes e identificação de ruídos no ambiente. Os sensores magnéticos são utilizados para medir a intensidade, direção e sentido do campo magnético terrestre. Os sensores de localização são utilizados para informar a posição geográfica de um objeto.

Além dos sensores físicos (ou de *hardware*) acima mencionados, existem também os sensores virtuais (ou de *software*) que podem ser utilizados para extrair outros tipos de

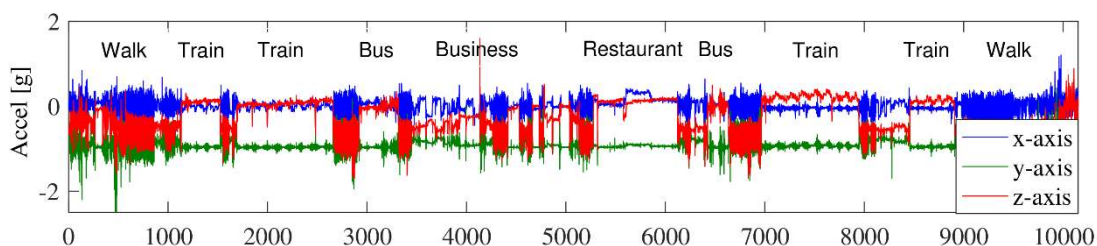
informações de contexto relacionadas ao usuário. No caso dos smartphones, os sensores virtuais podem ser vistos como os aplicativos. Por exemplo, aplicativos de jogos, bancos e utilitários podem fornecer informações importantes sobre o perfil e as atividades dos usuários. Portanto, a combinação dos dados extraídos tanto dos sensores de *hardware* quanto dos sensores de *software* abre portas para o surgimento de novas oportunidades para o desenvolvimento de soluções inovadoras de RAH para smartphones.

Neste contexto, esta tese de doutorado é baseada somente no uso dos sensores inerciais acelerômetro e giroscópio, incluindo o magnetômetro devido ao foco da pesquisa ser somente o reconhecimento das atividades de movimento (ou físicas) dos usuários. Dessa forma, esta seção apresenta detalhes de como funcionam esses sensores e o formato de dados gerados por cada um deles.

### 2.3.1 Acelerômetro

O acelerômetro é caracterizado por detectar o início e o fim de um movimento através da medição da força da aceleração aplicada sobre o dispositivo. Matematicamente, os dados podem ser visualizados em um plano tridimensional ao longo dos eixos  $x$ ,  $y$  e  $z$ . Dessa forma, os dados brutos do acelerômetro são representados por um conjunto de três vetores  $acc_i = (x_i, y_i, z_i)$ , onde  $i = (1, 2, 3, \dots, n)$ . A frequência da coleta de dados é medida em *Hertz* (Hz), ou seja, a quantidade de amostras que o sensor gera no intervalo de 1 segundo. Por exemplo, se o sensor gerar 50 amostras por segundo, então a frequência da coleta dos dados é 50Hz. Nos smartphones, a frequência pode variar entre 1Hz e 200Hz.

Além de medir a aceleração, o acelerômetro também é capaz de medir a força da gravidade aplicada sobre o dispositivo. Tecnicamente, a gravidade representa a força aplicada sobre o dispositivo ao longo dos eixos  $x$ ,  $y$  e  $z$ . Existe a possibilidade de eliminar a informação de gravidade dos dados do acelerômetro por meio de técnicas matemáticas. A exclusão da gravidade transforma esses dados tornando-os lineares. Além disso, os dados lineares de um acelerômetro também podem ser extraídos de sensor específico denominada de *acelerômetro linear*.



**Figura 2.1 Exemplos de dados extraídos de um acelerômetro com padrões de diferentes atividades executadas ao longo do tempo. Fonte [96].**

O *acelerômetro* é o sensor mais utilizado no RAH para smartphones devido sua capacidade de representar bem as atividades de movimento dos usuários. A Figura 2.1 mostra um exemplo de dados coletados de um acelerômetro. O eixo  $x$  do gráfico mostra a sequência ordenada de coleta das informações das coordenadas  $x$ ,  $y$  e  $z$  ao longo do tempo. O eixo  $y$  mostra a amplitude dos valores das coordenadas. Como pode ser observado, o padrão do trecho de sinal contido no intervalo de 0 a 1000 no eixo  $x$  é diferente do padrão do trecho de sinal contido no intervalo entre 7000 a 8000. Essa diferença é devido a execução de duas atividades diferentes sendo a primeira a atividades “andar” e a segunda a atividade “andar de trem”. A mesma intuição é válida para as demais atividades mostradas na parte superior da figura.

Basicamente, o objetivo das soluções de HAR é reconhecer esses padrões e classificá-los de acordo com as atividades correspondentes.

### 2.3.2 Giroscópio

O sensor giroscópio é caracterizado por medir a direção do dispositivo por meio da taxa de rotação medida em rad/s (radianos por segundo). Matematicamente, os dados também podem ser visualizados em um plano tridimensional ao longo dos eixos  $x$ ,  $y$  e  $z$ . Dessa forma, os dados brutos do giroscópio são representados por um conjunto de vetores  $gyr_i = (x_i, y_i, z_i)$ , onde  $i = (1, 2, 3, \dots, n)$ . Nesse sensor, os dados brutos também são coletados em forma de frequência medidas em *Hertz*, similar à coleta de dados do acelerômetro. Cada eixo do giroscópio possui os seguintes significados:

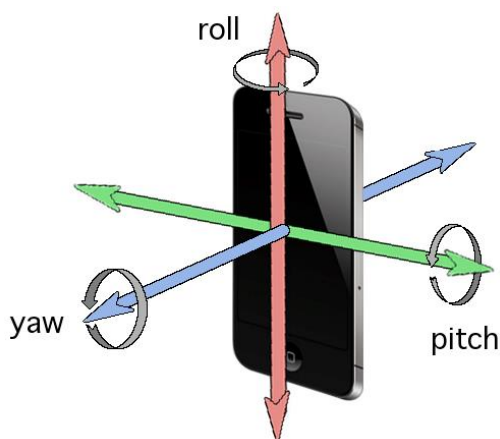


Figura 2.2 Exemplo das rotações em cada eixo do smartphone. Fonte: Google.

- **Eixo x (roll):** mede rotação horizontal do dispositivo, ou seja, mudança na direção do dispositivo (esquerda e direita).
- **Eixo y (pitch):** mede a rotação da inclinação do dispositivo.
- **Eixo z (yaw):** mede a rotação vertical do dispositivo, ou seja, mudança na orientação do dispositivo (retrato e paisagem).

Em resumo, o *giroscópio* mede a mudança angular do dispositivo ao longo tempo. A Figura 2.2 mostra um exemplo das rotações dos eixos.

### 2.3.3 Magnetômetro

O sensor *magnetômetro* é caracterizado por medir a intensidade, a direção e o sentido do campo magnético terrestre. Este sensor é frequentemente utilizado como bússola fornecendo informações de orientação de direção (norte/sul) onde é indicado o ponto cardinal norte do globo terrestre. Além disso, este sensor também é utilizado para detectar metais em torno do dispositivo. Matematicamente, os dados também podem ser visualizados em um plano tridimensional ao longo dos eixos  $x$ ,  $y$  e  $z$ . Dessa forma, os dados brutos do giroscópio são representados por um conjunto de vetores  $gyr_i = (x_i, y_i, z_i)$ , onde  $i = (1, 2, 3, \dots, n)$ .

## 2.4. Abordagens utilizadas para reconhecer as Atividades Humanas

Para reconhecer as atividades humanas, a computação dispõe de vários métodos computacionais que atuam em etapas diferentes de processamento e manipulação de dados. Portanto, o processo de reconhecimento de atividades humanas é fortemente similar a um sistema de reconhecimento

de padrões de propósito geral. As metodologias comumente utilizadas possuem um conjunto de etapas bem definidas que abrangem à coleta e segmentação dos dados, extração e seleção de características e treinamento de modelos de classificação de atividades [97] [22]. Esse processo envolve um conjunto de transformações nos dados brutos extraídos dos sensores com objetivo de gerar modelos eficientes de classificação de atividades humanas. Essa metodologia tem sido base para o emprego de técnicas de inteligência artificial no âmbito dos algoritmos de aprendizagem de máquina *offline*, representados pelos algoritmos de aprendizagem de máquina rasos e profundos, e os algoritmos *online* que são baseados nos algoritmos de análise de fluxo contínuo de dados. A Figura 2.3 fornece uma visão geral das abordagens utilizadas em HAR.



**Figura 2.3** Tipos de abordagens empregada nas soluções de HAR. Fonte: autor.

O primeiro bloco trata dos algoritmos de aprendizagem de máquina *offline* rasos (e.g. SVM, Árvore de Decisão, *Naive Bayes* e KNN). Tais algoritmos têm sido amplamente utilizados na literatura desde o surgimento dos smartphones. Essa abordagem concentra a maior parte das soluções de HAR e o principal foco dos estudos tem sido a etapa de extração de características. Essa etapa é responsável por transformar os sinais dos sensores inerciais em dados que possam ser compreendidos pelos algoritmos de aprendizagem de máquina. Esses dados são compostos por um conjunto de atributos (ou características) capazes de discriminar as atividades (ver Seção 3.3). Por exemplo, cada atividade pode ser representada por uma média e variância padrão ao longo de um intervalo de dados representado por um segmento do sinal.

O segundo bloco trata dos algoritmos de aprendizagem *offline* profundos (e.g. CNN, RNN, RBM, SAE, DFN e DBM) [24]. A principal vantagem desses algoritmos é a capacidade de gerar as características de forma automática em contraste com os algoritmos rasos. Além disso, esses algoritmos são capazes de otimizar as características para cada cenário, ou seja, cada base de dados de treino tende a ter um conjunto específico de características que melhor representa a distribuição dos dados. Essas características são totalmente desconhecidas. A teoria por trás disso afirma que o conhecimento humano é limitado de forma que tais atributos dificilmente seriam definidos por um especialista [24]. Por essa razão, os algoritmos de aprendizagem profunda são altamente eficientes e possuem as melhores taxas de acurácia disponíveis na literatura, sendo assim, considerados o estado da arte dos modelos de classificação gerados por algoritmos *offline*.



O terceiro bloco trata dos algoritmos *online* baseados no fluxo contínuo de dados [35]. As principais vantagens dessa abordagem são: 1) capacidade de atuar em tempo real (*online*) por meio da adaptação constante dos modelos de classificação; 2) os modelos de classificação são especializados no usuário; e 3) os algoritmos utilizam poucos recursos computacionais de memória e processamento devido a não necessidade de armazenar histórico de dados. A metodologia utilizada por esses algoritmos é diferente da metodologia padrão utilizada pelos algoritmos *offline*. Nesse caso, as etapas da metodologia consistem em: 1) armazenar dados recentes e descartar dados antigos utilizando um espaço de memória curta (e.g. *buffer*); 2) extrair informações fundamentais dos dados para compor o modelo de classificação; 3) monitorar a evolução dos dados ao longo do tempo com o propósito de atualizar o modelo de classificação quando ocorrer alteração nos dados; e 4) descobrir e classificar novas atividades que vão surgindo ao longo do tempo. Os algoritmos *online* também têm sido pouco utilizados pelas soluções de HAR.

## 2.5. Desafios e Oportunidades

O processo de HAR tem sido bastante explorado devido às oportunidades e desafios que a área evidencia. Tais oportunidades vêm emergindo de alguns problemas que ainda não foram resolvidos na área de HAR no contexto dos smartphones. Alguns desafios [98] encontrados nas soluções de HAR para os smartphones com sensores inerciais têm sido herdados de um contexto similar explorado pelos sensores vestíveis no corpo humano (*body-worn*). Entretanto, os smartphones possuem problemas peculiares como uma maior probabilidade de imersão em ambientes naturais, uma vez que os usuários carregam os smartphones para todos os lugares. Sendo assim, a maioria desses problemas vem acompanhada das limitações das próprias tecnologias de *hardware* ou *software* empregadas nas soluções atuais. Esta seção apresenta alguns problemas relacionados aos smartphones e, portanto, oportunidades e desafios a serem superados.

**Diversidade de ambientes e atividades:** O primeiro problema está relacionado com a variedade de ambientes e da diversidade de atividades que os humanos são capazes de executar [23]. Além disso, cada usuário possui um *fingerpint* biológico [28] o qual pode ser refletido no padrão de suas atividades. Por este motivo, diversos trabalhos baseados em autenticação através de atividades humanas têm sido desenvolvidos na literatura [99] [100]. Neste contexto, existem diversos desafios e limitações que necessitam ser superados como reconhecimento de atividades individuais [26], a capacidade de detecção de novas atividades [89] e reconhecimento de atividades concorrentes [16]. Como solução para esse problema, estudos podem ser realizados em direção ao emprego de técnicas de detecção de novidades com foco em algoritmos não-supervisionados e/ou semi-supervisionados. O objetivo é evitar um extensivo treinamento de modelos de classificação com bases de dados limitadas como ocorre com os algoritmos de classificação rasos [89].

**Aprendizado em tempo real:** o segundo problema está relacionado com o reconhecimento das atividades humanas em tempo real [78]. A maioria das soluções existentes gera os modelos de classificação de forma *off-line*, onde os modelos de classificação são gerados em um servidor e enviados para os smartphones. A desvantagem dessa abordagem é que o envio de dados de um smartphone para um servidor consome muita energia da bateria o que torna essa estratégia inviável na prática. Para resolver este problema, Shoaib et al. [101] apresentam um conjunto de trabalhos que realizam o treinamento dos modelos de classificação

no próprio smartphone utilizando técnicas de baixo custo. No entanto, os resultados mostram que o custo de processamento dos dados e de consumo de energia ainda é considerado alto para os smartphones. A solução para este problema seria a utilização de algoritmos de aprendizagem *online*, uma vez que os modelos de classificação são gerados ao longo do tempo sem a necessidade do armazenamento de um histórico de dados e o baixo uso de memória. Dessa forma, a carga de processamento é reduzida consideravelmente quando novas atividades surgirem [89].

**Modelos flexíveis para atividades complexas:** o terceiro problema está relacionado com a falta de modelos eficientes capazes de reconhecer atividades complexas. A principal dificuldade é que essas atividades contêm informações semânticas e de contexto que não são tratadas nas soluções estudadas até agora. Além disso, os dados dos sensores possuem uma variabilidade de características diferentes que dificultam a combinação dos dados de múltiplos sensores. Nessa perspectiva, duas soluções podem ser aplicadas nesse problema. A primeira está vinculada ao avanço da área de fusão de dados de sensores e a segunda é o emprego de informações de contexto relacionadas a localização, conexão de redes, aplicações, entre outras. Além disso, mais estudos relacionados com detecção de novas atividades e aprendizagem ativa ajudariam a resolver este problema [89] [102].

**Escalabilidade das soluções de HAR:** O quarto problema está relacionado com a escalabilidade das soluções de HAR [16]. Até agora, as pesquisas têm sido realizadas com uma quantidade limitada de dados e atividades. Essa prática dificulta a generalização das soluções desenvolvidas devido à diversidade de comportamentos que os indivíduos possuem durante a execução de suas atividades. Como solução para este problema, três estratégias podem ser utilizadas: 1) Emprego de técnicas de transferência de aprendizagem entre modelos [20]; 2) Uso de dados provenientes do sensoriamento de multidões [103]; e 3) Emprego de algoritmos *online* para geração de modelos personalizados para cada usuário [89].

A solução proposta por esta tese de doutorado tem como objetivo solucionar todos os problemas dos desafios mencionados acima com o desenvolvimento do algoritmo NOHAR (Capítulo 6).

## 2.6. Aplicações Práticas

A seguir são listadas as áreas em que o processo de descoberta, reconhecimento e previsão de atividades humanas podem ser úteis.

**Saúde:** com o aumento da popularidade de tecnologias assistidas, a área da saúde vem se tornando um dos principais focos da área de HAR [53]. Exemplos de aplicações como detecção de quedas [3], monitoramento de pessoas idosas [4] e prevenção de doenças [5] têm motivado o desenvolvimento de soluções que avaliam e intervêm no estado de saúde dos usuários por meio da análise de comportamento. Além disso, o monitoramento de atividades físicas na perspectiva de cuidados com a saúde do usuário, também tem sido bastante explorado pelas soluções de HAR. Aplicações relacionadas às atividades aeróbicas [29], estimativa de gasto de energia corporal [104], práticas de esportes [105] e detecção de passos [73] são exemplos de soluções desenvolvidas nesse cenário.

**Smartphones e ambientes inteligentes:** a combinação entre os dados dos smartphones e de ambientes inteligentes é outro campo de aplicação explorado. Soluções como monitoramento de atividades domésticas [6], economia de energia elétrica [7], modelagem de

ambientes domóticos [8] e medição do nível de poluição de um ambiente [9] são alguns exemplos das soluções desenvolvidas para esta área. Assim, este cenário tem gerado soluções que oferecem comodidade ao usuário na execução de tarefas rotineiras, especialmente, em ambientes de casas inteligentes.

**Segurança:** a área de segurança é um campo de aplicação clássico da área de HAR por causa da difusão das câmeras como sensores visuais. Soluções de monitoramento de atividades [10], detecção de anomalias em multidões [11] e rastreamento de objetos [12] são exemplos de soluções desenvolvidas para este cenário. As soluções desta área têm sido bastante utilizadas em sistemas de defesas e departamentos de investigação. Outra aplicação que está sendo explorada recentemente é a autenticação do usuário baseada em suas atividades. Nessa perspectiva, cada usuário possui um padrão específico para executar suas atividades físicas. Tal característica tem despertado interesse para o desenvolvimento de soluções de autenticação baseadas no comportamento. Neste sentido, alguns trabalhos [99] [100] têm desenvolvido soluções de autenticação com base na análise de dados dos sensores inercias. Assim, tais soluções têm reforçado os métodos tradicionais de autenticação como uso de PIN (*Personal Identification Number*), senhas e *fingerprint*.

**Redes sociais:** a área de redes sociais tem sido um campo de aplicação bastante utilizado no reconhecimento de atividades em grupo. Soluções voltadas para detecção de comunidades [106], identificação de relacionamentos [107] e sensoriamento de grupos [108] são exemplos de soluções desenvolvidas neste cenário. As soluções dessa área têm sido utilizadas para melhorar o entendimento do comportamento humano em comunidade.

**Localização:** o HAR baseado em localização é outro campo de aplicação bastante explorado na literatura. Soluções de logística [13], transporte [14] e navegação de pedestres [15] são exemplos de soluções desenvolvidas para este cenário. Soluções desta área têm sido utilizadas para melhorar a locomoção do trânsito e pedestres, além de recomendar locais de interesse ao usuário.

Além dessas aplicações existem outras citadas por Shoaib et al. [109] que englobam marketing, entretenimento, assinatura biométrica pessoal e monitoramento de atividades de trabalhadores em processos de manufatura na indústria. Como pode ser visto, a área de HAR possui um amplo campo de aplicação em cenários reais. Assim, novas pesquisas nesta área vão auxiliar no aprimoramento das soluções existentes e no desenvolvimento de mais soluções inovadoras que venham melhorar cada vez mais os processos rotineiros dos usuários e comunidades.

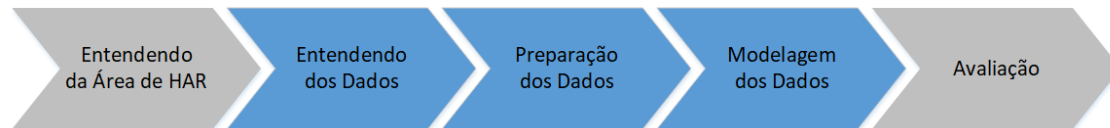
## 2.7. Considerações Finais

Este capítulo apresentou uma visão geral da área de HAR baseada em smartphones incluindo conceitos de atividades humanas, um histórico que mostra a evolução da área de HAR por meio de uma linha do tempo e uma descrição dos sensores inerciais utilizados para reconhecer as atividades físicas dos usuários. Além disso, este capítulo mostrou mais detalhes sobre os desafios e oportunidades discutidos na Seção 1.1, além de destacar as principais aplicações práticas da área de HAR foram apresentadas. Mais detalhes sobre os conceitos de HAR podem ser obtidos nos trabalhos [13] [16] [17] [18].

As discussões apresentadas neste capítulo juntamente com a descrição da metodologia de HAR descrita no próximo capítulo contêm parte do conteúdo do survey submetido para a revista Sensors.

# CAPÍTULO 3

## METODOLOGIA PADRÃO DE HAR E TRABALHOS RELACIONADOS



Este capítulo apresenta detalhes das etapas da metodologia padrão comumente empregada nas soluções de HAR. Na etapa de coleta de dados, o capítulo aborda como os dados dos sensores inerciais são tratados em termos de frequência da coleta, posição e orientação do smartphone no corpo do usuário. Na etapa de preparação dos dados, o capítulo descreve um conjunto de transformações nos dados necessárias para atuação dos algoritmos de modelagem e inferência de dados. Nesse sentido, o capítulo aborda as principais técnicas de segmentação de dados, o processo de extração de características e fusão de dados. Na etapa de modelagem, o capítulo apresenta os algoritmos de classificação rasos e profundos comumente utilizados para gerar modelos de classificação das atividades humanas. Por fim, é apresentado as principais técnicas de fusão de dados utilizadas na área de HAR.

Para apresentação dos trabalhos relacionados, este capítulo adotou uma estratégia diferente da convencional. À medida que as etapas da metodologia são explanadas, nós apresentamos várias tabelas que mostram uma lista de trabalhos que utilizaram as técnicas descritas em cada etapa.

### 3.1. Metodologia Padrão para Análise de Dados

A Figura 3.1 mostra uma visão geral das etapas comumente utilizadas pelas metodologias de HAR. Cada uma delas serão estudadas com mais detalhes nos próximos capítulos. A etapa inicial é referente à coleta de dados brutos dos sensores dos smartphones como, por exemplo, acelerômetro e giroscópio. Para isso, um conjunto de parâmetros como tipo, tempo e frequência da coleta de dados, bem como, a posição e a orientação do smartphone no corpo do usuário devem ser levados em consideração. A maioria dos smartphones comumente utilizados na etapa de coleta de dados possui sistemas operacionais embarcados como *Android*, *iOS* e *Symbia*.

A próxima etapa é referente à segmentação dos dados. A segmentação é responsável por organizar os dados brutos em grupos, de tal forma que cada um deles compartilhe as mesmas características. Na perspectiva dos sensores inerciais, essa organização é feita por meio da divisão da base de dados em segmentos sequenciais conhecidos como janelas de tempo. Alguns trabalhos [110] [29] [79] acrescentam um passo anterior referente ao pré-processamento dos dados brutos, com o propósito de minimizar os ruídos causados por anomalias relacionadas às condições ambientais, movimentos e mudanças de comportamento do usuário durante a coleta de dados. Dessa forma, as técnicas de eliminação de ruídos mais utilizadas são os filtros *Lowpass* [110], média móvel [29] e o de *Kalman* [79].

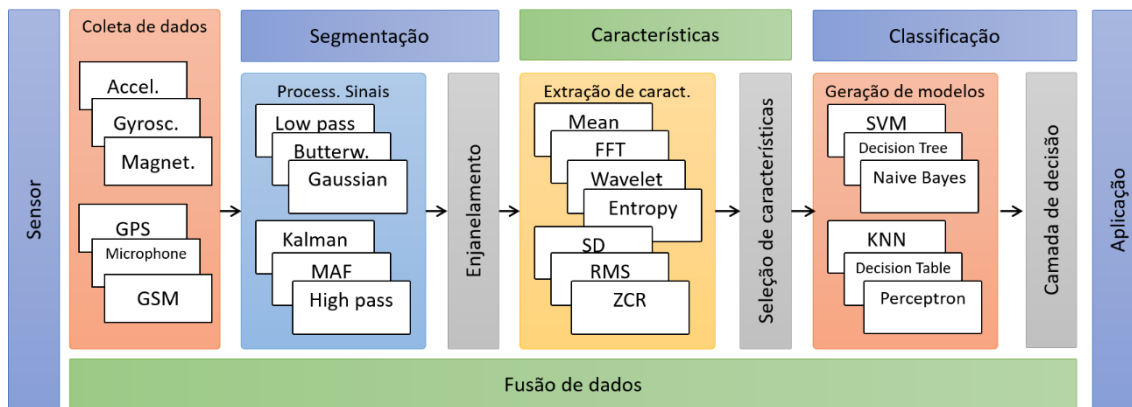


Figura 3.1 Metodologia geral para soluções de HAR. Fonte: autor.

A etapa de extração de características consiste em um processo de transformação dos dados realizada em cima dos dados segmentados. O objetivo é que cada atividade seja representada por informações úteis capazes de diferenciar as atividades umas das outras. Nesta etapa são utilizadas as características dos domínios de tempo, frequência e discreto. Cada um deles será apresentado no Capítulo 3. Além disso, técnicas de seleção de características são aplicadas com o propósito de eliminar aquelas que são irrelevantes para os modelos de classificação. A última etapa é referente à construção dos modelos de classificação e à inferência das atividades humanas. Os modelos de classificação são gerados com base nos algoritmos de aprendizado de máquina rasos e profundos.

A etapa de fusão de dados é um processo transversal utilizado para combinar dados de múltiplas fontes. Esse aspecto transversal acontece porque os dados podem ser combinados em qualquer uma das etapas descritas acima. Por exemplo, na etapa de coleta e segmentação dos dados podem ser aplicadas as técnicas de filtros de sinal como o de Kalman, na etapa de extração de características pode ser utilizada a estratégia de concatenação de vetores de características dos dados dos sensores e, por fim, na etapa de classificação pode ser utilizada um conjunto (*ensemble*) de classificadores. Todas essas estratégias têm o propósito de reforçar a precisão da classificação das atividades.

## 3.2. Coleta de Dados

A Figura 3.2 mostra a coleta e armazenamento dos dados dos sensores inerciais, acelerômetro e giroscópio. Os dados são organizados em ordem cronológica numa tabela composta por linhas e colunas. Cada coluna da tabela representa o tempo, as coordenadas  $x, y$  e  $z$  e a atividade correspondente.

Para que os dados sejam representativos e consigam gerar bons modelos de classificação, os seguintes elementos devem ser levados em consideração: tipo, tempo, taxa de frequência, posição e orientação do smartphone no corpo do usuário. Lockhart e Weiss [111] propuseram três tipos de coletas de dados baseados no nível de naturalidade com que os dados são coletados:

- I. **Natural:** quando os usuários executam suas atividades diárias normalmente sem intervenção no seu comportamento pela aplicação.
- II. **Semi-natural:** quando os usuários executam suas atividades diárias normalmente, porém, o usuário é obrigado a executar, pelo menos uma vez, as atividades previstas

nos experimentos. Em outras palavras, o usuário deve assegurar que todas as atividades relacionadas ao estudo tenham sido executadas.

- III. **Laboratório:** quando os usuários executam as atividades sistematicamente em um ambiente controlado com passos previamente definidos.

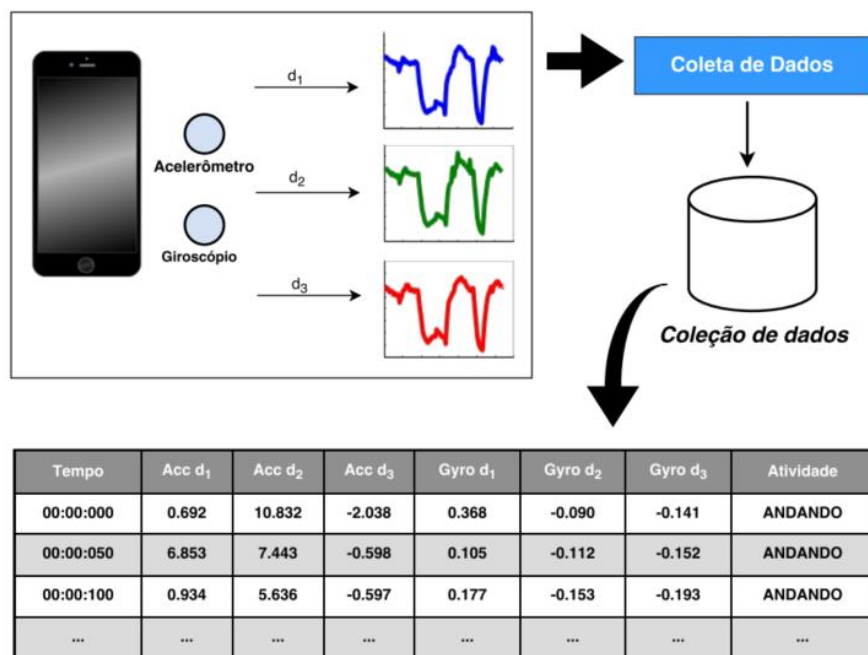


Figura 3.2 Exemplo de coleta e armazenamento de dados dos sensores inerciais. Fonte: autor.

A revisão da literatura mostra que a coleta de dados realizada em laboratórios tende a gerar modelos de classificação mais precisos, devido às atividades pertencentes aos estudos em questão serem previamente definidas. Entretanto, os modelos gerados com esse tipo de base de dados perdem acurácia quando aplicados em contextos reais devido à diversidade de comportamento dos usuários. Por outro lado, os modelos gerados com bases de dados naturais tendem a ser mais genéricos podendo ser aplicados em grupos de pessoas com comportamentos similares. A Tabela 3.1 apresenta alguns estudos divididos pelo tipo de coleta de dados.

Tabela 3.1 Lista de trabalhos separados por tipos de coleta de dados.

Tipo de Coleta	Trabalhos
Natural	[67] [29] [76] [31] [78] [64] [112] [83] [113] [114] [115] [116] [86] [54]
Seminatural	[72] [57] [81] [117] [118] [119] [120]
Laboratório	[73] [30] [74] [67] [75] [110] [79] [64] [121] [53] [82] [84] [122] [115] [123] [124] [27] [92]

A frequência da coleta dos dados é, também, um fator importante que deve ser levado em consideração pelas soluções de HAR, pois a quantidade de amostras no intervalo de um segundo contém informações relevantes dos movimentos do corpo humano [101]. Na literatura, as frequências utilizadas para extrair dados dos sensores inerciais de smartphones variam em torno de 1 Hz a 200 Hz. Khusainov et al. [23] realizaram um estudo com intuito de descobrir qual a taxa de frequência que é suficiente para representar bem os movimentos do corpo

humano. Os resultados mostraram que a frequência de 20 Hz contém dados suficientes capaz de discriminar os movimentos. A Tabela 3.2 mostra uma lista de trabalhos que realizaram estudos em diversas faixas de frequências.

**Tabela 3.2 Lista de trabalhos separados por frequência.**

<b>Frequência (Hz)</b>	<b>Trabalhos</b>
1-20	[72] [30] [76] [79] [78] [84] [123] [64] [81] [121] [82] [83] [84] [118] [86] [124]
30-80	[29] [78] [57] [114] [116] [86] [119] [125] [110] [80] [121] [53] [113] [122] [120] [86] [54] [92]
100-200	[74] [31] [63] [84] [115] [117] [86]

A posição do smartphone no corpo do usuário é outro fator que influencia bastante na qualidade dos dados coletados e na precisão dos modelos de classificação. Por exemplo, os dados coletados com o smartphone posicionado na cintura do usuário produzem padrões de sinais diferentes do smartphone posicionado na mão do usuário [80] [86]. A Tabela 3.3 mostra uma lista de trabalhos separados por posições do smartphone no corpo.

**Tabela 3.3 Lista de trabalhos separados por posição.**

<b>Posição</b>	<b>Trabalhos</b>
Qualquer	[72] [73] [30] [75] [110] [31] [115] [86] [123] [119] [54]
Cintura	[67] [79] [57] [80] [121] [112] [122] [120] [124] [91] [92]
Bolso da calça	[74] [67] [57] [64] [81] [80] [121] [82] [122] [114] [120] [116] [118] [54] [91] [27]
Cordão no pescoço	[67]
Mão	[63] [122] [118] [91] [27]
Braço	[57] [91]
Peito	[57] [80] [121] [120] [116] [54]
Mochila	[57] [118]

A criação de modelos genéricos para o reconhecimento de atividades com o smartphone localizado em diferentes posições do corpo do usuário ainda é um desafio a ser superado. Estudos realizados com o smartphone em diferentes posições mostram que a cintura é a melhor posição para reconhecer atividades físicas, visto que a origem dos movimentos do corpo humano ocorre nesta região [80]. Além disso, alguns trabalhos descritos em [101] têm empregado soluções que são independentes da posição do smartphone no corpo humano. De modo resumido, essas soluções têm focado na geração de classificadores treinados com dados dos smartphones localizados em todas as posições relevantes do corpo do usuário.

A orientação do smartphone no corpo do usuário também é outro fator que influencia na precisão dos modelos de classificação. O que torna a dependência ou a independência de orientação dos modelos de classificação são os tipos de características utilizadas no processo de treinamento desses modelos. Por exemplo, as características extraídas do sinal de magnitude são consideradas características independentes de orientação, pois os seus valores não alteram com as mudanças de orientação do dispositivo. A Tabela 3.4 apresenta uma lista de trabalhos que observaram a orientação dos smartphones para gerar os modelos de classificação.



**Tabela 3.4 Lista de trabalhos separados por orientação.**

<b>Orientação</b>	<b>Trabalhos</b>
Dependente	[67] [75] [31] [79] [78] [57] [64] [81] [112] [53] [82] [83] [84] [125] [124] [54] [27] [92]
Independente	[72] [73] [30] [74] [29] [110] [76] [63] [80] [121] [113] [122] [114] [115] [116] [117] [118] [123] [119] [120] [92]

Por fim, a diversidade dos dados permite uma maior generalização dos modelos de classificação. Portanto, um grande número de pessoas é um fator essencial para a obtenção de dados representativos. A revisão da literatura mostra que as quantidades de pessoas não ultrapassaram 50 indivíduos, devido principalmente à dificuldade de persuasão dos usuários em fornecer suas informações pessoais. Para diversificar os cenários é necessário obter dados de pessoas de diferentes faixas etárias com diferentes níveis de saúde e locomoção, dentre outros fatores.

### 3.2.1 Bases de Dados

No contexto de HAR para smartphones, a literatura apresenta dois tipos de base de dados comumente utilizadas nos experimentos para validação das soluções de HAR. O primeiro trata das bases de dados geradas pelos próprios autores e o segundo trata de bases de dados públicas. Os primeiros trabalhos de HAR baseados em smartphones utilizavam bases de dados próprias nos seus experimentos. A desvantagem desse cenário é que, além da dificuldade na coleta dos dados por parte dos autores, as comparações entre as soluções eram prejudicadas devido à impossibilidade de reprodução dos experimentos num mesmo cenário. Somente após 2011, com a publicação da base de dados pública WISDM [81], a comparação entre diversas soluções de HAR tornou-se mais viáveis. A partir daí diversas outras bases de dados públicas surgiram na literatura. A Tabela 3.5 apresenta uma lista das principais bases de dados de sensores inerciais utilizadas pelos trabalhos de pesquisa. É importante ressaltar que existem outras bases de dados listadas em [24] e [126], porém, as demais não são frequentemente utilizadas ou pertencem a outros domínios. Nessa perspectiva, o Apêndice A descreve com detalhes as bases de dados utilizadas nos experimentos conduzidos por este trabalho.

**Tabela 3.5 Lista de bases de dados públicas. A – Acelerômetro, G - Giroscópio e M – Magnetômetro.**

<b>Base de dados</b>	<b>Frequencia</b>	<b>Sensores</b>	<b>Usuários</b>	<b>Referência</b>
OPPORTUNITY	30 Hz	A, G, M	12	[127]
UCI-HAR Smartphone	50 Hz	A, G	30	[128]
PAMAP2	100 Hz	A, G, M	9	[129]
USC-HAD	100 Hz	A, G	14	[130]
WISDM e Actitracker	20 Hz	A	29	[81]
MHealth	50 Hz	A, G	10	[131]
Extra Sensory	40 Hz	A, G, M	60	[102]
Shoaib	50 Hz	A, G, M	10	[91]
UniMib Shar	50 Hz	A	30	[132]

### 3.3. Segmentação

A segmentação tem como propósito separar os dados em sub-grupos significativos que compartilham as mesmas características. No contexto dos sensores inerciais, os sub-grupos de dados são representados por segmentos de sinais em um determinado intervalo de tempo. O objetivo é que cada segmento contenha informações suficientes que possibilitem o reconhecimento de uma atividade humana em determinado momento, ou seja, a análise de dados deve ser feita exatamente durante um intervalo de tempo de execução de cada atividade.

Para alcançar esse objetivo, os dados são divididos em segmentos consecutivos de forma que cada um deles seja analisado de forma separada e sequencial. Este processo é conhecido como enjanelamento de séries temporais (ou janelas deslizantes). A segmentação baseada em janela deslizante é frequentemente utilizada na manipulação dos dados dos sensores inerciais e de áudio. Essa abordagem divide os dados em subsequências organizadas em intervalos de tempo.

Cook e Krishnan [133] e Lin et al. [134] definem janela deslizante como sendo derivada de uma sequência  $X = \{x_1, x_2, \dots, x_n\}$ , onde  $x$  representa o valor e  $n$  o  $n$ -ésimo valor da sequência. A janela de tempo é representada por uma subsequência  $X' = \{x_p, x_{p+1}, \dots, x_{p+w-1}\}$ , onde  $w$  representa o tamanho da janela de tempo e  $p$  representa uma posição arbitrária, tal que  $1 \leq p \leq n - w + 1$ , onde  $n$  representa o tamanho da sequência. No caso dos sensores inerciais, os dados são representados em um plano tridimensional ao longo dos eixos  $x_i, y_i, z_i$ , onde  $i = (1, 2, 3, \dots, n)$ .

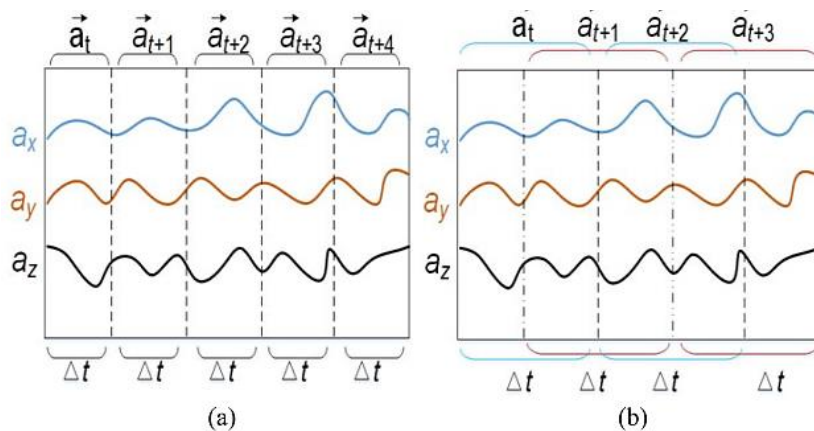


Figura 3.3 Exemplo de segmentação não-sobreposta (a) e segmentação com sobreposição (b). Fonte [93].

A segmentação baseada em janela deslizante pode ser manipulada de duas formas [23]: janelas sobrepostas e não-sobrepostas. As janelas não-sobrepostas são segmentos onde seus valores não possuem intersecção com os valores de outras janelas, ou seja,  $X_1 \cap X_2 = \emptyset$ . As janelas sobrepostas são segmentos representados por uma porcentagem que define quantas amostras da janela anterior compõem as amostras da janela posterior, ou seja,  $X_1 \cap X_2 \neq \emptyset$ . Por exemplo, dada uma janela de tempo com 100 amostras, 50% de sobreposição significa que 50 amostras da janela anterior farão parte do conjunto de amostras da próxima janela. A Figura 3.3 mostra um exemplo de segmentação baseada em janela deslizante.

No contexto de sensores inerciais, os tamanhos das janelas são medidos com base no intervalo de tempo e na frequência da coleta de dados. Além disso, as janelas podem possuir tamanhos fixos ou variáveis. A quantidade de amostras do tamanho fixo é definida com base no

tempo comumente medido em segundos. Estudos mostram que o tamanho ideal para as janelas fixas varia em torno de 2 a 5 segundos considerando uma frequência de 20Hz a 50 Hz [87] [92] [27]. Por outro lado, a quantidade de amostras das janelas de tamanho variável pode ser definida de acordo com, por exemplo, as mudanças na média e na variância do sinal. Porém, no melhor do nosso conhecimento, ainda não realizaram estudos com tamanhos de janelas variáveis. Todas as soluções encontradas até agora utilizam janelas de tempo de tamanho fixo. A Tabela 3.6 mostra uma lista de trabalhos com seus respectivos tamanhos de janelas de tempo utilizados.

**Tabela 3.6 Lista de trabalhos separados por tamanho de janelas.**

<b>Tamanho de Janela de tempo (seg)</b>	<b>Trabalhos</b>
< 1	[75] [31] [64] [80] [84] [120] [54]
1-5	[73] [74] [63] [57] [80] [53] [82] [84] [113] [122] [120] [115] [116] [117] [118] [86] [123] [54] [87] [92] [27]
7-60	[72] [29] [81] [53] [114] [119] [125] [27]

Para estudos relacionados com tamanhos variáveis de janelas de tempo, sugerimos emprego de uma técnica que detecta variações na média do sinal como a técnica Page Hinkley proposta por Sebastião et al. [135]. A intuição por trás dessa ideia é que as fronteiras das janelas de tempo podem ser exatamente a posição da mudança na média do sinal. Isso resolve o problema de que em uma determinada janela de tempo contenha dados misturados de mais de uma atividade.

Outro fator importante que influencia no processo de segmentação é a limpeza dos dados por meio da redução de ruídos dos sensores inerciais. Tais ruídos podem ser caracterizados por valores faltantes, valores incorretos e *outliers*. A eliminação dos ruídos pode ser feita por meio de técnicas específicas comumente utilizadas na área de processamento de sinal. Um levantamento feito no estado da arte mostra que as soluções de HAR baseadas em smartphones usam, principalmente, filtros como passa-baixa [121] [85] [54] [84] [86] [124] [54] (do inglês, *Low-pass*), Butterworth [110], Kalman [79] e Média Móvel (do inglês, *Moving Average Filter*) [86] [29] [63].

### 3.4. O que são características?

Intuitivamente, uma característica pode ser definida como um elemento dotado de alguma informação útil sobre o dado da qual ela representa. No contexto de HAR, esse conceito pode ser utilizado para representar os diferentes padrões de movimento das atividades físicas dos usuários. Por exemplo, a atividade “correr” exige maior esforço do corpo humano para gerar movimento comparado à atividade “andar”. Logo, a intensidade do esforço de cada uma delas é transferida para os sensores inerciais, influenciando diretamente na diferença entre a distribuição dos dados desses sensores. Portanto, é possível encontrar formas de destacar a diferença entre a atividade “andar” e “correr” utilizando, por exemplo, informações estatísticas dos dados. Dessa forma, a média e a variância dos dados, por exemplo, podem ser úteis para destacar a diferença entre essas duas atividades.

Neste contexto, a literatura classifica as características em diferentes domínios de representação. Cada domínio possui um conjunto de fórmulas específicas que extraem diferentes informações úteis dos sinais dos sensores inerciais. Os grupos de classificação de

domínios de características definidos pela literatura são [37]: domínio do tempo e o domínio da frequência. O domínio de tempo possui funções matemáticas utilizadas para extrair informações estatísticas dos sinais. O domínio de frequência possui funções matemáticas que capturam padrões repetitivos de sinais e que são, frequentemente, relacionados com a periodicidade natural das atividades.

### 3.4.1 Domínio do Tempo

As características baseadas no domínio de tempo são divididas em dois tipos de funções: as funções estatísticas e as não-estatísticas. As funções estatísticas envolvem cálculos como mínimo, máximo, média, desvio padrão, entre outras fórmulas. As funções não-estatísticas envolvem cálculos diversos como áreas, cálculo de *Bins Distribution* [136] [109], entre outros. Para facilitar o entendimento das fórmulas, considere a definição de janela de tempo descrita na seção 3.1. A Tabela 3.7 mostra o conjunto de fórmulas usadas na extração de características de tempo mapeadas na literatura. Todas são aplicadas nos eixos  $x_i, y_i$  e  $z_i$  dos sensores inerciais.

**Tabela 3.7 Time Features.**

Time Features	Formulas	Time Features	Formulas
Minimum	$Min(X)$	Standard deviation	$\sqrt{\frac{1}{n} \sum_{k=0}^n (x_k - Mean(X))^2}$
Maximum	$Max(X)$	Pearson coefficient	$\frac{\frac{1}{n} \sum_{k=0}^n (x_k - Media(X))^2}{\frac{1}{n} \sum_{k=0}^n x_k}$
Amplitude	$ Max(X) - Min(X) $	Mean Square	$\sqrt{\frac{1}{n} \sum_{k=0}^n x_k^2}$
Sum	$\sum_{k=0}^n x_k$	Absolute deviation mean	$\frac{1}{n} \sum_{k=0}^n  x_k - Mean(X) $
Absolute sum	$\sum_{k=0}^n  x_k $	Error square sum	$\sum_{k=0}^n (x_k - Mean(X))^2$
Energy	$\sum_{k=0}^n x_k^2$	Variance	$\frac{1}{n} \sum_{k=0}^n (x_k - Mean(X))^2$
Energy Log	$\sum_{k=0}^n \log(x_k^2)$	Skewness	$\frac{\frac{1}{n} \sum_{k=1}^n (x_k - Mean(X))^3}{(\frac{1}{n} \sum_{k=1}^n (x_k - Mean(X))^2)^{\frac{3}{2}}}$
Euclidian Norm	$\sqrt{\sum_{k=0}^n x_k^2}$	Kurtosis	$\frac{\frac{1}{n} \sum_{k=1}^n (x_k - Mean(X))^4}{(\frac{1}{n} \sum_{k=1}^n (x_k - Mean(X))^2)^3} - 3$
Power	$\frac{1}{n} \sum_{k=0}^n x_k^2$	Bin	$\cup_{i=1}^n \frac{i * (Max(X) - Min(X))}{n}$
Area	$\sum_{k=0}^n x_k + y_k + z_k$	Absolute signal magnitude mean	$\frac{1}{n} \sum_{k=0}^n ( x_k  +  y_k  +  z_k )$
Mean	$\frac{1}{n} \sum_{k=0}^n x_k$	Signal magnitude mean	$\frac{1}{n} \sum_{k=0}^n \sqrt{x_k^2 + y_k^2 + z_k^2}$
Absolute Mean	$\frac{1}{n} \sum_{k=0}^n  x_k $	Absolute area	$\sum_{k=0}^n  x_k  +  y_k  +  z_k $
Zero-crossing			$\frac{1}{2n} \sum_{k=0}^n  sign(x_k) - sign(x_{k-1}) $
Percentile			$P(p) = \left\lceil \frac{p}{100} * n \right\rceil, p = kth \text{ value}$
Correlation			$\frac{n(\sum_{k=1}^n x_k y_k) - (\sum_{k=1}^n x_k)(\sum_{k=1}^n y_k)}{\sqrt{n(\sum_{k=1}^n x_k^2) - (\sum_{k=1}^n x_k)^2} + \sqrt{n(\sum_{k=1}^n y_k^2) - (\sum_{k=1}^n y_k)^2}}$

Auto-correlation	$\frac{\sum_{k=1}^{n-1}(x_k - \text{Mean}(X))(x_{k+1} - \text{Mean}(X))}{\sum_{k=1}^n(x_k - \text{Mean}(X))^2}$
Cross-correlation	$\frac{\sum_{k=1}^n(x_k - \text{Mean}(X))(y_k - \text{Mean}(Y))}{\sqrt{\sum_{k=1}^n(x_k - \text{Mean}(X))^2 \sum_{k=1}^n(y_k - \text{Mean}(Y))^2}}$
Average Magnitude Difference function (AMDF)	$\frac{\frac{1}{N} \sum_{k=0}^{N-1}  x_k - x_{k+N} }{\text{Max}(X)}$

Dentre as características mencionadas acima, existem algumas características especiais que podem gerar outras novas características por meio de um processo de encadeamento de funções matemáticas. Por exemplo, a característica magnitude, descrita logo abaixo, pode ser combinada com outras características como média e variância e gerar novas características a partir dessa combinação [117]. O mesmo ocorre com as características baseadas nos componentes verticais e horizontais dos sinais [80] [122]. Além disso, os sinais gerados por essas características especiais possuem tamanhos iguais ao tamanho dos sinais originais, enquanto as demais características geram sinais compactados com tamanhos iguais a quantidade de janelas de tempo utilizadas no sinal. Em outras palavras, essas características funcionam como técnicas de fusão de dados de coordenadas, uma vez que as coordenadas  $x_i$ ,  $y_i$  e  $z_i$  são transformadas em apenas uma coordenada.

Por essas razões, este trabalho classifica as características com este tipo de comportamento como características de baixo nível, onde o processo de extração é realizado nos dados brutos. Para entender melhor como funcionam essas características, Sousa et al. [92] apresentam detalhes de cada uma delas no contexto de HAR para smartphones.

A magnitude de sinal é uma característica independente de orientação do smartphone no corpo do usuário, extraída de dados de sensores com múltiplas dimensões. Segundo Khusainov et al. [23], o objetivo da magnitude é avaliar o grau de intensidade de movimento com base em limiares derivados dos picos de aceleração. Isso é possível devido à magnitude destacar as variações de sinais causadas pela fusão de valores entre diferentes coordenadas dos sensores inerciais. A Equação 3.1 mostra como calcular a magnitude de sinal em dados extraídos de sensores inerciais cujas coordenadas são  $x$ ,  $y$  e  $z$ :

$$M(X_i) = \sqrt{x^2_i + y^2_i + z^2_i} \quad (\text{Equação 3.1})$$

Da mesma forma, as características baseadas nos componentes verticais ( $v_i$ ) e horizontais ( $h_i$ ) também são consideradas características independentes de orientação do smartphone no corpo do usuário [29] [115]. Essas características também aplicam uma fusão dos valores das coordenadas  $x$ ,  $y$  e  $z$  dos sensores inerciais. Formalmente, a Equação 3.2 apresenta a fórmula para o cálculo da componente vertical:

$$v_i = \vec{a}_i \cdot \hat{g} \quad (\text{Equação 3.2})$$

onde  $\vec{a}_i = (x_i, y_i, z_i)$ ,  $1 \leq i \leq m$ ,  $i$  representa os valores das amostras em uma janela de tamanho  $m$ . O valor de  $\hat{g}$  é um vetor unitário que representa a gravidade contida no sinal. A gravidade  $\hat{g}$  pode ser extraída de acordo com a Equação 3.3:

$$\hat{g} = \frac{(x, y, z)}{\|(x, y, z)\|} \quad (\text{Equação 3.3})$$

onde  $\bar{x}, \bar{y}, \bar{z}$  representa a média dos valores de cada coordenada contidos em uma janela de tempo e  $\|(\bar{x}, \bar{y}, \bar{z})\|$  representa a norma<sup>1</sup> do vetor de valores das coordenadas. As características baseadas nos componentes horizontais ( $h_i$ ) são derivadas das características baseadas nos componentes verticais ( $v_i$ ). Assim, o componente horizontal é calculado pela fórmula da Equação 3.4:

$$h_i = \|\bar{a}_i - v^{proj}_k\| \quad (\text{Equação 3.4})$$

onde,  $v^{proj}_k$  significa uma projeção do componente vertical calculada pelo produto do escalar componente vertical pelo vetor unitário da gravidade do sinal (Equação 3.5).

$$v^{proj}_k = v_i \hat{g} \quad (\text{Equação 3.5})$$

O Apêndice B mostra uma lista de trabalhos que utilizaram as características abordadas acima.

### 3.4.2 Domínio da Frequência

As características do domínio de frequência apresentam uma alternativa de análise de sinal baseada no espectro de frequência dos valores de determinada janela de tempo. As características da Tabela 3.8 são calculadas com base nas características de baixo nível *Fast Fourier Transform* (FFT) ou *Wavelet*. Para facilitar o entendimento das fórmulas, considere a definição de janela de tempo descrita na seção 3.1. A Tabela 3.8 mostra o conjunto de fórmulas usadas na extração de características de frequência mapeadas na literatura. Todas são aplicadas nos eixos  $x_i, y_i$  e  $z_i$  dos sensores inerciais.

**Tabela 3.8 Frequency Features.**

Frequency Features	Fórmulas
Energy	$\sum_{k=1}^n x^2$
Energy normalized	$\sum_{k=1}^n \left(\frac{x_k}{\sum_{k=1}^n x_k}\right)^2$
Power	$\frac{1}{n} \sum_{k=1}^n x^2$
Centroid	$\frac{\sum_{k=1}^n x_k * n}{\sum_{k=1}^n x_k}$
Entropy	$-\sum_{k=1}^n F_n \log F_n$ , tal que $F_n = \frac{x_k}{\sum_{k=1}^n x_k}$
Dominant component	$\frac{1}{n} \sum_{k=1}^n x_k$
Coefficientes Sum	$\sum_{k=1}^n x_k$

As características do domínio de frequência descritas na Tabela 3.8 dependem, estritamente, das características de baixo nível transformadas de Fourier e Wavelet. Ambas transformadas consistem em uma ferramenta matemática que realiza transição entre variáveis ao longo do tempo para variáveis de frequência. Assim, o sinal é decomposto em um conjunto de valores, reais e imaginários, que representam componentes de ondas denominados de frequências. No contexto de HAR, as transformadas são úteis para representar padrões repetitivos de sinais em termos de frequência.

Para calcular a transformada de Fourier é utilizado um algoritmo eficiente utilizado no cálculo da transformada de Fourier discreta (*Discrete Fourier Transform* - DFT), denominado

<sup>1</sup> A norma de um vetor é calculada por  $\|v\| = \sqrt{a^2 + b^2}$ .

de transformada rápida de Fourier (do inglês, *Fast Fourier Transform* - FFT) cuja fórmula é representada pela Equação 3.6:

$$\text{FFT}(X_i) = \sum_{k=1}^{n-1} x_k e^{-2\pi i j \frac{k}{n}} \quad (\text{Equação 3.6})$$

onde  $x_k$  é um valor da sequência na posição  $k$  e  $e^{-2\pi i j \frac{k}{n}}$  representa a  $n$ ésima raiz primitiva de cada unidade de  $x_k$ . Mais detalhes sobre a raiz primitiva pode ser encontrada em [137]. O cálculo da transformada Wavelet é similar ao cálculo da transformada de Fourier, a diferença é que os valores de um intervalo são representados em termos de bases ortogonais. Existem várias formas de calcular a transformada de Wavelet observando valores contínuos e discretos. Entretanto, no contexto de HAR, a literatura utiliza uma variação da transformada de Wavelet denominada de Haar Wavelet. Esta forma é representada pela fórmula básica de cálculo de Wavelets apresentada por Strang [138].

Além disso, algumas características pertencentes ao domínio do tempo podem ser adaptadas para o domínio da frequência como pico da amplitude da frequência dos coeficientes, média dos coeficientes, área dos coeficientes, entre outras. Visto que os coeficientes reais e imaginários também formam vetores é possível extrair informações estatísticas sobre eles.

As características derivadas das transformadas de Fourier e Wavelet são consideradas dependentes de orientação do smartphone no corpo do usuário. O Apêndice B mostra uma lista de trabalhos que utilizaram as características abordadas acima.

### 3.5. Extração de Características

De modo geral, a extração de características corresponde a um processo de transformação dos dados realizada em cima dos dados segmentados. No contexto dos sensores inerciais, esse processo é necessário porque os dados brutos, representados por séries temporais, não são adequados para serem utilizados pelos algoritmos de aprendizagem de máquina rasos [133] [139]. Nesse sentido, esta seção apresenta detalhes de como as características dos domínios de tempo e frequência são extraídas de uma série temporal. Além disso, esta seção aborda aspectos relacionados à redução de dimensionalidade dos dados nas bases de dados geradas após a etapa de extração de características.

As características do domínio do tempo e da frequência são extraídas da mesma maneira, i.e., ambas são provenientes de processamentos realizados sobre as janelas de tempo definidas na etapa de segmentação. Por este motivo, o tamanho e a taxa de sobreposição das janelas de tempo implicam diretamente na qualidade das características. O processo de extração de características gera uma nova base de dados no padrão adequado para ser utilizada na etapa de treinamento e geração dos modelos de classificação das atividades. A Figura 3.4 mostra um exemplo visual de como ocorre a extração de características no sinal do acelerômetro. Neste caso, o sinal é segmentado em três janelas de tempo de tamanhos fixos e, em seguida, como exemplo são extraídas duas características de cada janela de tempo. Tais características são representadas pelo somatório dos valores absolutos (*Absolute Sum* - AS) e pela média da raiz quadrada de cada valor (*Root Mean Square* - RMS). A Tabela 3.9 mostra a distribuição de estudos separados pelos domínios de características.

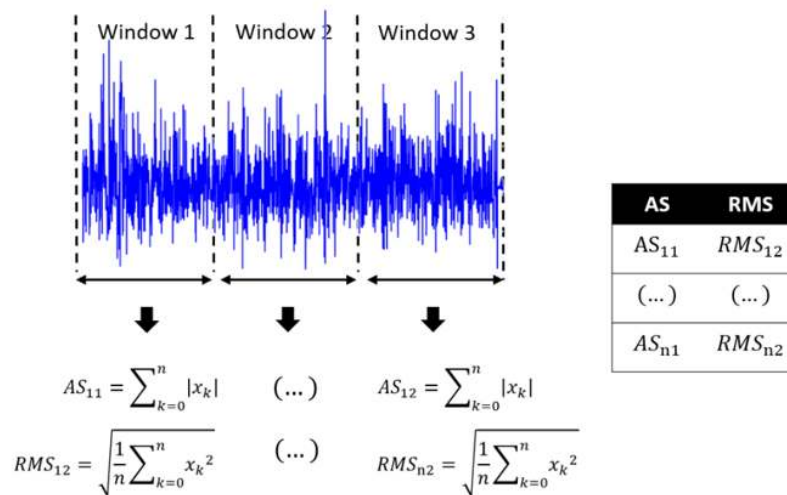


Figura 3.4 Exemplo de extração de características e geração de uma base de dados secundária. Fonte: autor.

Alguns dados importantes indicam que a maioria dos trabalhos voltados para HAR em smartphones com sensores inerciais utiliza as características baseadas no domínio de tempo [111] [86]. O motivo da ampla utilização dessa abordagem é que, em geral, as características de domínio do tempo possuem um custo computacional menor quando comparadas com as características do domínio de frequência [101] [118]. Em contrapartida, as características do domínio de frequência conseguem representar melhor as informações de contexto em termos de padrões de sinais.

Tabela 3.9 Lista de trabalhos separados por características.

Domínios	Trabalhos
Tempo	[72] [30] [74] [67] [29] [110] [76] [31] [78] [63] [57] [64] [81] [121] [112] [53] [82] [83] [84] [113] [122] [114] [120] [115] [116] [117] [118] [86] [123] [119] [125] [124] [54]
Frequência	[73] [74] [29] [78] [57] [112] [113] [122] [115] [116] [118] [123] [125]

As características de baixo nível descritas na Seção 4.2 podem ser divididas com base na orientação do smartphone no corpo do usuário, ou seja, orientação modo retrato (*portrait*) e orientação modo paisagem (*landscape*). A característica de magnitude, por exemplo, é considerada uma característica independente de orientação, enquanto que as características FFT e *Wavelet* são consideradas dependentes de orientação [116]. Em consequência, todas as novas características derivadas de qualquer característica de baixo nível, herdam a classificação de dependência ou independência de orientação. Por outro lado, as características do domínio do tempo e de frequência, quando aplicadas isoladamente são todas dependentes de orientação. A Tabela 3.10 mostra uma lista de trabalhos que utilizaram as características de baixo nível.

Tabela 3.10 Lista de Trabalhos que utilizam as características de baixo nível.

Características	Trabalhos
Baixo Nível	[73] [74] [67] [29] [110] [76] [63] [57] [80] [121] [84] [113] [122] [114] [120] [115] [116] [117] [118] [123] [119]



### 3.5.1 Redução de Dimensionalidade dos Dados

A dimensionalidade é atribuída à quantidade de características contidas em uma base de dados, sendo que, cada característica representa uma dimensão no espaço. Desse modo, um grande espaço de características evidencia dois problemas. O primeiro está relacionado com o custo de processamento dos dados e o segundo com a precisão dos modelos de classificação gerados na etapa de aprendizagem. Além disso, existe o problema da maldição da dimensionalidade que pode propiciar altas taxas de erro em um classificador [140]. Na prática, a maldição da dimensionalidade implica que para um dado tamanho de amostras, existe um número máximo de características a partir do qual o desempenho do classificador irá degradar, ao invés de melhorar. A solução para tal problema é reduzir a dimensão através da seleção de características ou de métodos de redução de dimensionalidade.

O processo de redução de dimensionalidade dos dados consiste na remoção de características irrelevantes com o propósito de melhorar a precisão dos modelos de classificação. Por esta razão, Khusainov et al. [23] afirmam que a escolha das características são mais importantes que escolha dos algoritmos de classificação, uma vez que a baixa qualidade das características pode impactar negativamente na precisão de qualquer modelo gerado pelos algoritmos de aprendizagem de máquina.

A literatura de redução de dimensionalidade de dados é bem vasta [140], porém, poucos trabalhos utilizaram tais técnicas no contexto de HAR baseada em smartphone. Mesmo assim é possível identificar dois tipos de técnicas utilizadas nesse contexto. O primeiro trata de técnicas que atuam após a etapa de extração de características e o segundo trata de técnicas que atuam durante a etapa de extração de características.

O primeiro tipo é caracterizado pelas técnicas de seleção de características. Basicamente, esses conjuntos de técnicas selecionam as características mais representativas dentre todas disponíveis na base de dados. No contexto de HAR, podemos citar alguns exemplos utilizados nas soluções propostas por Khusainov et al. [23] e Khan [141]. Ambas as soluções utilizaram as técnicas baseada na análise do ganho de informação das características (*Info Gain*) e no nível de correlação entre elas (*Correlation-based Feature Selection*). Tais técnicas analisam o impacto que cada característica causa no desempenho dos modelos de classificação. Assim, características que possuem baixo ganho de informação ou são altamente correlacionadas entre si, podem ser descartadas da base de dados.

Enquanto que as técnicas de seleção de características definem, de fato, o sub-conjunto de características que melhor discriminam as atividades humanas, os métodos que atuam durante a etapa de extração de características combinam as características de tal forma que, além de reduzir a dimensionalidade dos dados, são geradas novas características que possuem menor variância intra-classe e maior variância entre-classes com o propósito de aumentar a separabilidade das atividades antes de alimentar o classificador [141]. Neste contexto, as técnicas mais utilizadas no contexto de HAR são: o PCA (do inglês, *Principal Component Analysis*), LDA (do inglês, *Linear Discriminant Analysis*) e KDA (do inglês, *Kernel Discriminant Analysis*).

Recentemente, novos métodos de redução de dimensionalidade têm sido empregados no contexto de HAR. Por exemplo, Hassan et al. [142] utilizam o método Kernel PCA (KPCA) com um kernel estatístico que melhora o desempenho do PCA. Siddiqi et al. [143] utilizam o método não linear denominado de *Stepwise Linear Discriminant Analysis* (SWLDA) que seleciona características discriminantes por meio de métodos de regressão combinados com

técnica estatística F-test. Por fim, Guo e wang [144] utilizaram uma modificação do método LDA denominado de *Robust Linear Discriminant Analysis* (RLDA).

### 3.6. Treinamento e Classificação

Após o processamento dos dados nas etapas de segmentação e extração de características, o próximo passo é o emprego de algoritmos de classificação responsáveis por gerar os modelos de inferência das atividades humanas. Neste âmbito, os algoritmos de classificação são divididos em dois grupos. O primeiro trata dos algoritmos de aprendizagem de máquina rasos e o segundo dos algoritmos de aprendizagem profunda. Dessa forma, os modelos de inferência (ou classificação) são gerados por meio de um processo de treinamento dos algoritmos de classificação. Esses modelos são gerados a partir de um histórico de dados onde as amostras de atividades devem estar devidamente rotuladas com as atividades no qual se deseja classificar. Formalmente, o processo de treinamento desses modelos é definido da seguinte forma [20]:

Dada uma variável aleatória  $X$  pertencente a um plano  $n$ -dimensional. Essa variável corresponde a atributos, ou características, extraídas das sequências de eventos de sensores. Assim,  $X = \langle x_1, x_2, \dots, x_n \rangle$ , onde  $x$  representa uma característica e  $n$  a quantidade de característica. A variável  $X$  possui uma variável  $y$  associada que representa um atributo alvo ou classe da variável  $X$ . Assim,  $y = \langle y_1, y_2, \dots, y_n \rangle$ , onde  $y$  representa a classe e  $n$  a quantidade de classes. Dessa forma, um conjunto de treinamento  $L$  consiste de pares  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , onde  $(x_n, y_n)$  representa uma instância formada por um conjunto de atributos  $x^n$  e uma classe  $y^n$ . Por fim, um modelo de classificação é gerado de modo que uma instância não rotulada seja prevista para alguma instância contida na variável  $y$ .

Os modelos de classificação são categorizados da seguinte forma [28] [145]:

- **Impessoais ou genéricos:** os modelos são treinados com dados de um grupo de usuários e testados em outro grupo de usuários diferentes.
- **Pessoais ou específicos:** os modelos são treinados com dados de apenas um usuário e testado com o mesmo usuário.
- **Mistos:** os modelos são treinados usando toda a base de dados sem distinção entre os usuários.

Além disso, a literatura de HAR costuma avaliar os modelos de classificação com base na validação cruzada de 10 partições (do inglês, *Cross-Validation-10-folds*). Essa estratégia consiste em dividir os dados em 10 partes iguais, onde os modelos são gerados com 9 partes e testados com a parte restante. Isso se repete até que, individualmente, todas as partes sejam utilizadas como teste. A acurácia final consiste na média dos 10 modelos de classificação gerados em 10 rodadas de treinamento. A validação cruzada pode ser utilizada de duas formas:

- **Partições com dados aleatórios:** as partições são formadas com dados selecionados randomicamente e podem conter dados misturados de vários usuários.
- **Leave-one-subject-out:** As partições são formadas por dados de um único usuário. Isso garante que as partições utilizadas para teste não são vistas no treino.

Em geral, os trabalhos da literatura têm buscado desenvolver modelos impessoais com altas taxas de acurácia. Porém, gerar reconhecedores de atividades humanas que levem em consideração fatores como idade e saúde do usuário ainda é um desafio. Uma solução para esse

problema seria gerar modelos de classificação para cada perfil de usuário como crianças, adultos, pessoas com problemas de locomoção, entre outros.

### 3.6.1 Algoritmos de Aprendizagem Rasos

Os algoritmos de aprendizagem de máquina convencionais comumente utilizados para no reconhecimento de atividades físicas dos usuários são representados, por exemplo, pelos algoritmos Naïve Bayes, *Support Vector Machine* (SVM), redes neurais, KNN e as árvores de decisões. Esses algoritmos utilizam as características do domínio do tempo e da frequência, descritos na Seção 3.2, no processo de treinamento dos modelos de classificação. Não é objetivo desta tese fornecer informações teóricas de como cada um desses algoritmos funcionam, mais detalhes sobre cada um deles podem ser encontrado em [20]. A Tabela 3.11 apresenta uma lista de trabalhos que utilizam esses métodos e o Apêndice B mostra uma lista mais detalhada com outros algoritmos de classificação pouco utilizados pela literatura.

**Tabela 3.11 Lista de trabalhos separados por algoritmos de aprendizagem convencionais.**

Método	Trabalhos
Naïve Bayes	[72] [73] [74] [29] [78] [63] [53] [82] [83] [84] [116] [118] [123]
Decision Tree	[67] [29] [75] [76] [78] [57] [81] [53] [82] [113] [122] [116] [118] [123] [119]
SVM	[72] [29] [79] [78] [82] [115] [116] [118] [125] [54]
KNN	[29] [75] [63] [80] [82] [84] [114] [117] [118] [120]
Redes Neurais	[30] [81] [53] [86]

Os métodos baseados nas abordagens *boosting* e *bagging* também tem sido utilizados pelas soluções de HAR [20]. A abordagem *boosting* é uma técnica que combina múltiplos classificadores fracos com o propósito de produzir um único classificador forte. A intuição é que a precisão do modelo de classificação resultante da combinação dos classificadores é melhor que qualquer classificador fraco utilizado de forma separada. Para isso, a cada iteração o algoritmo recalcula a distribuição de dados e atribui pesos maiores a instâncias classificadas incorretamente. Assim, a cada nova iteração o algoritmo foca nas instâncias classificadas incorretamente na tentativa de melhorar o desempenho da classificação. Os algoritmos deste grupo mais conhecidos são o *AdaBoost* e o *Gradient Boosting*.

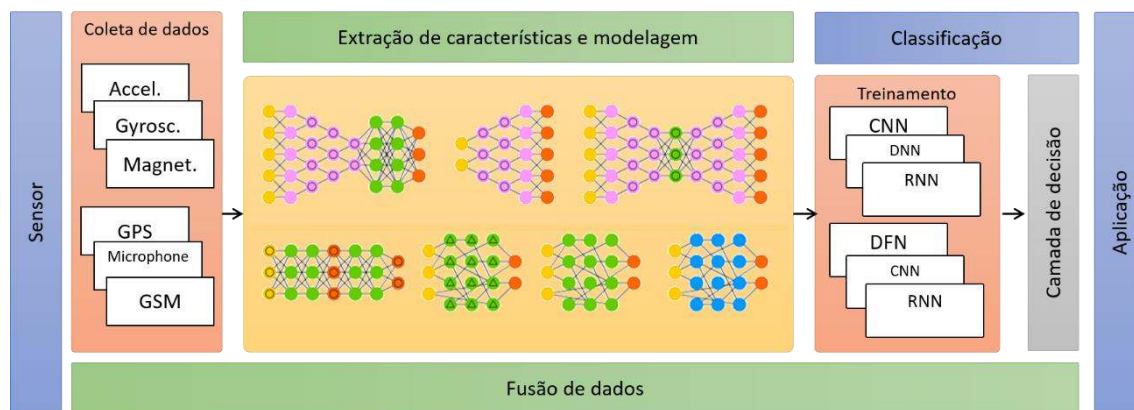
A abordagem *bagging* é um meta-algoritmo que também combina múltiplos modelos de classificação. A diferença entre as abordagens *bagging* e *boosting* é que a *bagging* produz várias réplicas da base de treino com tamanhos iguais e, então, um algoritmo de classificação é utilizado para treinar os modelos para cada réplica. No final, a classificação realizada por cada modelo é contabilizada, sendo que, e o modelo mais votado é utilizado para exibir o resultado final. O algoritmo *bagging* mais conhecido é o *Random Forest*.

### 3.6.2 Algoritmos de Aprendizagem Profundos

Do mesmo modo, os algoritmos de aprendizagem profunda estão sendo recentemente utilizados para reconhecer as atividades físicas dos usuários em smartphones. Neste âmbito, os algoritmos de classificação mais utilizados são: Deep fully-connected network (DFN), Convolutional neural network (CNN), Recurrent neural network (RNN), Long Short-Term Memory (LSTM), Stacked Autoencoder (SAE) e Restricted Boltzmann machine (RBM).

Diferente dos algoritmos de aprendizagem de máquina rasos, os algoritmos de aprendizagem profunda são capazes de gerar as características de forma automática [24]. Por

exemplo, as características dos domínios de tempo e frequência utilizadas pelos algoritmos rasos são extraídas de forma manual, visto que a definição dessas características depende do conhecimento humano. O problema é que o conhecimento humano é limitado [146] e a desvantagem é que, em alguns casos, a expertise humana nem sempre pode selecionar o melhor domínio de características para diferentes cenários. Outra desvantagem é que a abordagem rasa pode gerar características irrelevantes sendo necessário, portanto, a aplicação de métodos de redução de dimensionalidade e seleção de características.



**Figura 3.5** Conjunto de etapas baseadas na extração de características automáticas por meio da utilização de algoritmos de aprendizagem profunda. Fonte: autor.

Com o propósito de superar essas desvantagens, os algoritmos de aprendizagem profunda têm trazido um enorme benefício para a etapa de extração de características devido à habilidade de geração automática das características. Esses algoritmos são capazes de gerar características complexas que representam bem os dados e geram modelos de classificação eficientes para diferentes cenários. Por essa razão, os métodos de aprendizagem profunda são considerados o estado da arte em áreas como reconhecimento de objetos em imagens e processamento natural de linguagem [147].

Para ilustrar a diferença entre as abordagens rasa e profunda, a Figura 3.1 mostra o conjunto de etapas comumente utilizadas pelas abordagens rasas, com destaque para as etapas de segmentação e extração de características. A Figura 3.5 mostra o conjunto de etapas comumente utilizada pela abordagem baseada em algoritmos de aprendizagem profunda, onde as características são implicitamente geradas nas camadas escondidas das redes neurais, durante a fase de treinamento e construção dos modelos de classificação. As demais etapas são compartilhadas entre as duas abordagens. Não é foco desta tese apresentar os detalhes das arquiteturas de redes neurais, mais informações podem ser obtidas em [147].

**Tabela 3.12** Lista de trabalhos separados por algoritmos de aprendizagem profunda.

Métodos	Trabalhos
SAE	[94] [148] [149]
RBM	[150] [151] [152] [153] [154] [155] [156] [157] [158] [88]
CNN	[159] [160] [161] [162] [163] [164] [165] [166] [167] [168] [155] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178] [179] [180] [149] [181]
RNN	[182] [183] [164] [184] [185] [175] [178]
DFN	[164] [175] [186] [187] [188]
DBN	[189]
LSTM	[190]

A Tabela 3.12 apresenta uma lista de trabalhos que utilizaram esses métodos. Não é objetivo desta tese fornecer informações teóricas de como cada um desses algoritmos funcionam, mais detalhes sobre cada um deles pode ser encontrado em [147].

### 3.6.3 Métricas de Avaliação

O desempenho de um determinado modelo de classificação é avaliado por um conjunto de métricas que indicam o quão confiável é o modelo no processo de HAR. As principais métricas de avaliação comumente utilizadas na literatura de HAR baseada em smartphone são [133]: acurácia, sensibilidade, especificidade, precisão, revocação e *f-measure*.

A acurácia é a métrica mais comum utilizada para avaliar os modelos de classificação. No contexto de HAR, a acurácia é calculada pela divisão entre a quantidade das atividades corretamente classificadas,  $c$ , e a quantidade total de atividades  $n$ . A fórmula da acurácia é mostrada pela Equação 3.7.

$$Acuracia = \frac{c}{n} \quad (\text{Equação 3.8})$$

A acurácia passa uma ideia geral sobre os modelos de classificação. Porém, essa métrica trata as classes de uma base de dados como igualmente importantes. Isso torna a acurácia uma métrica ineficiente em bases de dados desbalanceadas. Para resolver esse problema existem outras métricas que avaliam as classes separadamente como a sensibilidade e a especificidade. A sensibilidade analisa a taxa de acerto dos verdadeiros positivos (VP) em relação aos falsos negativos (FN) para cada classe. A fórmula apresentada na Equação 3.9 mostra como calcula a sensibilidade:

$$sensibilidade = \frac{VP}{VP+FN} \quad (\text{Equação 3.9})$$

onde  $VP$  significa os verdadeiros positivos e  $FN$  significa os falsos negativos. Ao contrário, a especificidade analisa a taxa de acerto dos verdadeiros negativos (do inglês, *True Negative*) em relação aos falsos positivos (do inglês, *False Positive*) para cada classe. A fórmula apresentada na Equação 3.10 mostra como calcula a especificidade:

$$especificidade = \frac{VN}{VN+FP} \quad (\text{Equação 3.10})$$

onde  $VN$  significa os verdadeiros negativos e  $FP$  significa os falsos positivos. Da mesma forma ocorre com as métricas precisão, revocação e *f-measure*. A precisão analisa a taxa de acerto dos verdadeiros positivos em relação aos falsos positivos, enquanto que a revocação analisa a taxa de acerto dos verdadeiros positivos em relação aos falsos negativos. As fórmulas das Equações 3.11 e 3.12 mostram como calcula a métrica precisão e revocação.

$$precisao = \frac{VP}{VP+FP} \quad (\text{Equação 3.11})$$

$$revocacao = \frac{VP}{VP+FN} \quad (\text{Equação 3.12})$$

A métrica *f-measure* trata de uma pontuação resultante da combinação dos valores da precisão e revocação. A ideia dessa métrica é fornecer um valor genérico que represente essas duas métricas. A fórmula da Equação 3.13 mostra como calcula o *f-measure*:

$$f - measure = \frac{(1 + \beta)^2 * revocacao * precisao}{\beta^2 * (revocacao + precisao)} \quad (\text{Equação 3.13})$$

onde,  $\beta$  é um coeficiente de peso que comumente é atribuído o valor 1. A literatura costuma utilizar somente a acurácia para medir o desempenho dos modelos de HAR.

### 3.7. Fusão de Dados

A etapa fusão de dados corresponde ao processo de integração de múltiplos dados e/ou conhecimentos que representam um objeto do mundo real de forma consistente, precisa e útil [191]. As técnicas de fusão de dados são utilizadas com o propósito de melhorar a consistência dos dados e auxiliar na extração de informações cada vez mais completas de forma que um determinado conhecimento seja reforçado pela agregação de dados de múltiplas fontes.

Saeedi [15] e Tsinganos e Skodras [192] apresentam três níveis em que podem ser aplicadas as técnicas de fusão de dados. O primeiro nível pode ocorrer diretamente nos dados brutos dos sensores durante a etapa de segmentação. O segundo nível pode ocorrer na etapa de extração de características por meio da concatenação de vetores de características de múltiplas fontes. Por fim, o terceiro nível pode ocorrer na camada de decisão por meio da combinação dos resultados de vários classificadores. Um olhar atento na literatura de HAR para smartphones mostra que as técnicas de fusão de dados são mais aplicadas no contexto do segundo e terceiro nível. Nessa perspectiva, Vaizman et al. [102] e Köping et al. [95] apresentam três estratégias de fusão de dados no contexto de HAR para smartphones que envolvem apenas os dois últimos níveis, nesse trabalho [95] denominado de *Early Fusion* e *Late Fusion*.

A primeira estratégia (*Early Fusion*) trata da concatenação dos vetores de características de dois ou mais sensores em um único vetor. A segunda estratégia (*Late Fusion*) é baseada no método *Late Fusion using Average Probability* (LFA). O LFA utiliza uma heurística simples baseada no resultado de vários classificadores (*ensemble*). O LFA atribui pesos iguais para cada sensor na tentativa de eliminar a influência de sensores irrelevantes. A terceira estratégia é baseada no método *Late Fusion using Learned Weights* (LFL). O LFL atribui pesos diferentes para cada sensor na tentativa de descobrir sensores que reconhecem algumas atividades melhores que outros.

Embora Vaizman et al. [102] tenham proposto as estratégias baseadas no LFA e LFL, a maioria dos trabalhos aplicam a primeira estratégia baseada na concatenação das características. Dessa forma, para complementar essa estratégia, algumas soluções utilizam algoritmos de redução de dimensionalidade como PCA, LDA e KDA para auxiliar no processo de fusão de dados. Basicamente, esses algoritmos são utilizados na etapa final do processo de fusão de dados na tentativa de melhorar a fronteira de decisão entre as classes compostas pelo conjunto de características concatenadas. A seguir são apresentados os principais trabalhos que desenvolveram soluções utilizando fusão de dados.

Shoaib et al. [91] desenvolveram uma solução utilizando os dados dos sensores inerciais acelerômetro, acelerômetro linear, giroscópio e magnetômetro. Todos eles foram localizados em várias posições do corpo do usuário. A solução foi aplicada em três cenários. O primeiro cenário avaliou os modelos de classificação com o smartphone localizado em apenas uma posição do corpo do usuário (cintura). O segundo cenário avaliou os modelos de classificação com o smartphone localizado nas demais posições do corpo como no braço, pulso e bolso da calça. O terceiro cenário avaliou modelos de classificação específicos de um mesmo usuário.

Guiry et al. [85] desenvolveram uma solução para smartphones e *smartwatches* utilizando os dados dos sensores acelerômetro, giroscópio, magnetômetro, luz, pressão e GPS.

A etapa de fusão de dados foi dividida em 3 passos. O primeiro trata da aplicação da técnica de interpolação linear com o propósito de sincronizar os dados extraídos com diferentes frequências, pois os dados do acelerômetro, giroscópio, magnetômetro e pressão foram coletados à uma frequência de 100Hz, 27Hz, 25Hz e 5Hz, respectivamente. O segundo trata da concatenação dos vetores de características de todas as fontes de dados. Por fim, o terceiro trata da aplicação da técnica PCA para reduzir a dimensionalidade dos dados.

Khan et al. [54] apresentam uma solução para smartphones utilizando os dados dos sensores acelerômetro, pressão e microfone localizados em várias posições do corpo do usuário. Após o processo de concatenação das características, a etapa de fusão de dados conta com uma análise detalhada do impacto dos métodos de redução de dimensionalidade PCA, LDA e KDA na acurácia dos modelos de classificação. A análise concluiu que o método KDA com o kernel RBF (Gaussian) obtém os melhores resultados.

Outros trabalhos como o de Vepakomma et al. [186] combinam dados de sensores inerciais com sensores de ambientes inteligentes. Todos os trabalhos citados acima foram desenvolvidos utilizando os métodos de extração de características convencionais. No contexto de aprendizagem profunda, alguns trabalhos [174] [142] combinaram os dados dos sensores inerciais acelerômetro e giroscópio na própria arquitetura da rede neural sem qualquer pré-processamento extra no que diz respeito à fusão de dados.

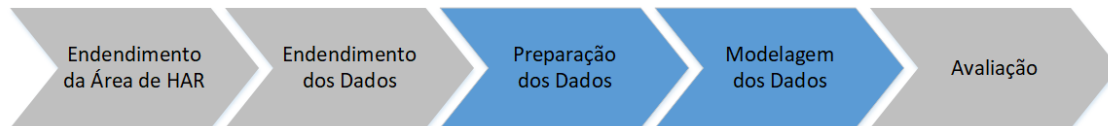
No contexto de hardware, as técnicas de fusão de dados também estão sendo implementadas na própria unidade de sensoriamento. Bancroft e Lachapelle [193], por exemplo, propuseram uma solução de fusão de dados para múltiplos chips IMU (*Inertial Measurement Unit*). Além disso, smartphones modernos possuem chips IMU com implementações embutidas de fusão de dados. Dessa forma, as extrações dos sinais não requerem qualquer pré-processamento, tal como, orientação, gravidade e aceleração linear. Neste ponto, as características de tempo e frequência podem ser diretamente calculadas.

### 3.8 Considerações Finais

Este capítulo apresentou os trabalhos relacionados e as etapas da metodologia comumente empregada em HAR, incluindo: 1) coleta de dados dos sensores inerciais; 2) a segmentação dos dados em janelas de tempo; 3) extração de características nos domínios do tempo e frequência; 4) fusão de dados de múltiplos sensores ou múltiplas coordenadas dos sensores inerciais; e 5) treinamento e classificação das atividades humanas usando algoritmos rasos e profundos. Mais detalhes sobre os trabalhos citados neste capítulo podem ser obtidos nas Tabelas do Apêndice B. Este capítulo finaliza o conteúdo que mostra uma visão mais aprimorada da área de HAR no contexto dos smartphones.

# CAPÍTULO 4

## REPRESENTAÇÃO SIMBÓLICA DOS DADOS



Este capítulo apresenta conceitos relacionados aos algoritmos de representação simbólica responsáveis por extrair características no domínio discreto. Em seguida, são apresentados os algoritmos de classificação simbólicos focados em manipular dados discretos. Tais algoritmos incluem SAX-VSM, BOSS-Model, BOSS-VS e WEASEL. Por fim, este capítulo apresenta várias estratégias de manipulação de séries temporais multidimensionais que podem ser utilizadas para tratar os dados de sensores inerciais, inclusive novas propostas adicionais provenientes desta tese de doutorado.

### 4.1. Características do Domínio Discreto

As características do domínio de tempo e frequência são representadas por valores contínuos extraídos por funções estatísticas e matemáticas aplicadas sobre um conjunto de segmentos de uma série temporal. Tais funções podem ser usadas para gerar numerosos atributos (e.g. média, desvio padrão e variância) a partir dos sinais de dados gerados pelos sensores, que juntos, são capazes de representar e discriminar uma classe ou atividade humana. Por outro lado, a representação dos dados no domínio discreto possui, como característica, um único atributo representado por uma sequência de valores simbólicos. Cada símbolo representa um intervalo de valores contínuos contidos em um determinado segmento do sinal. A Figura 4.1 mostra um exemplo de um segmento composto por 140 amostras que, quando discretizado, é representado pela palavra (ou símbolo) “*cbbccccbcdcbabca*”. Portanto, as características do domínio discreto são representadas por um conjunto de palavras, relacionados entre si, capazes de discriminar uma classe ou atividade humana a partir dos sinais extraídos dos sensores inerciais.

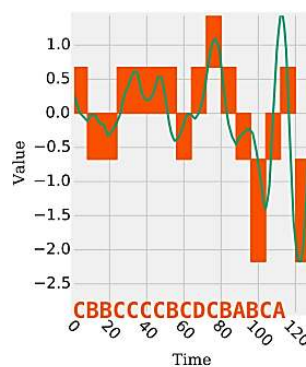


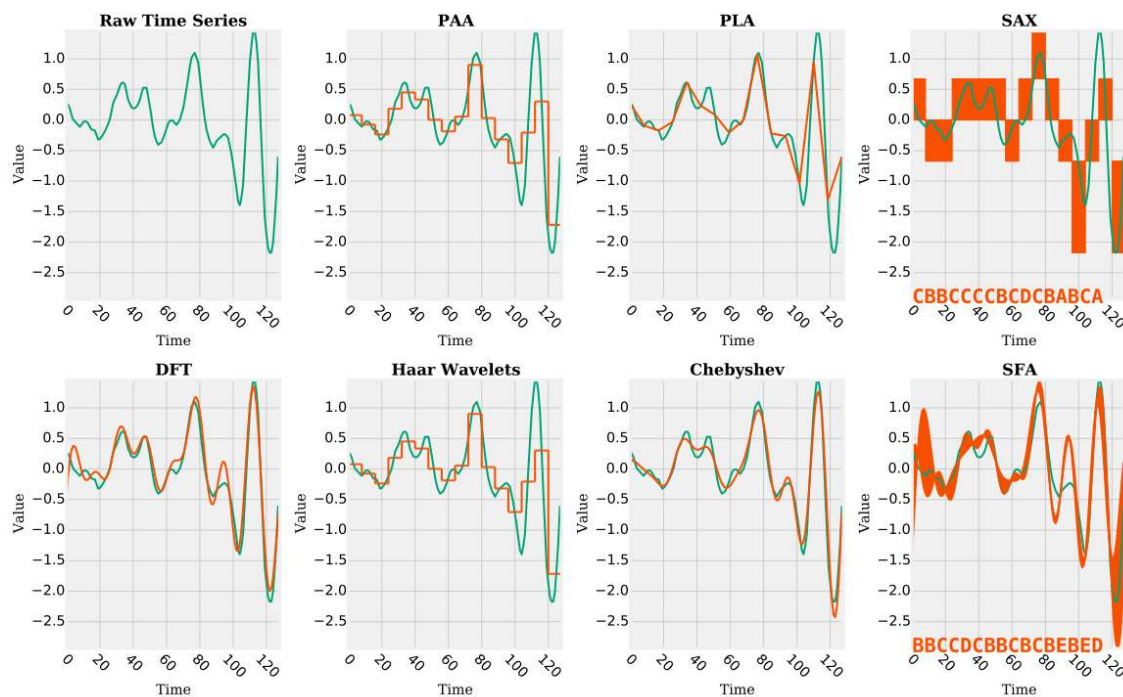
Figura 4.1 Exemplo de uma série temporal discretizada. Adaptado de [194].



#### 4.1.1 Extração de Características

Os algoritmos de representação simbólica transformam as séries temporais em uma representação compacta representada por uma sequência de símbolos que podem assumir um valor numérico ou discreto. Neste contexto, a literatura apresenta dois grupos de algoritmos que podem gerar características para o domínio discreto [39]. O primeiro grupo trata dos algoritmos de aproximação numérica incluindo *Discrete Fourier Transform* (DFT) [138], *Discrete Wavelet Transform* (DWT) [138], *Chebyshev Polynomials* (CP) [195], *Piecewise Linear Approximation* (PLA) [196], *Piecewise Aggregate Approximation* (PAA) [197] e *Adaptive Piecewise Constant Approximation* (APCA) [198]. O segundo grupo trata dos algoritmos de representação simbólica incluindo o *Symbolic Aggregate Approximation* (SAX) [38] e o *Symbolic Fourier Approximation* (SFA) [39]. A Seção 3.5.2 detalha como funcionam esses algoritmos. A Figura 4.2 mostra um exemplo visual contendo as diversas variações das características do domínio discreto utilizando os métodos citados acima.

Esta pesquisa foca nos algoritmos do segundo grupo por dois motivos. Primeiro, esses algoritmos são considerados estado da arte dos algoritmos de representação simbólica atualmente. Segundo, todos os algoritmos do segundo grupo são baseados em algum algoritmo do primeiro grupo. Por exemplo, o SAX e SFA definem os padrões de símbolos com base no PAA e nos coeficientes do DFT, respectivamente. Além disso, os símbolos utilizados por esses algoritmos são representados em forma de palavras formadas por um alfabeto de tamanho limitado. Basicamente, uma palavra representa uma aproximação numérica de uma subsequência composta por valores contínuos ordenados.



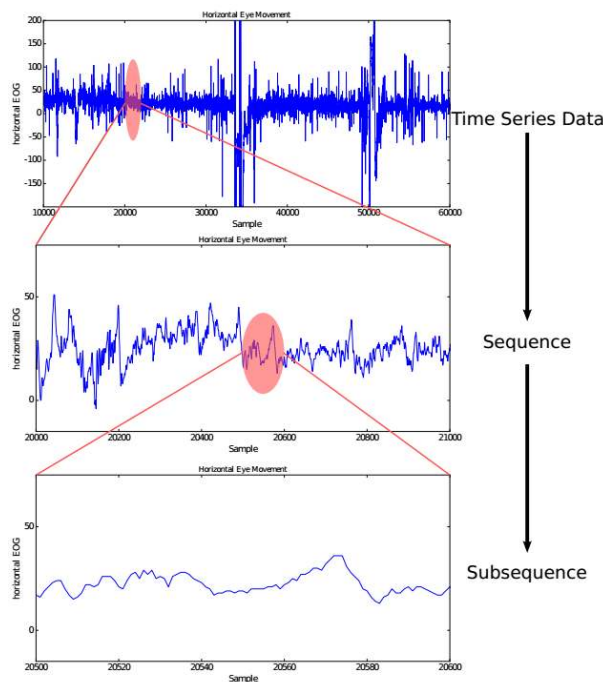
**Figura 4.2 Exemplos visuais dos resultados dos principais algoritmos de discretização identificados na literatura. Fonte [194].**

Embora existam riscos de perda de informação no processo de discretização, existe uma prova matemática, demonstrada por Lin et al. [199] e Schäfer e Högvist [39], que assegura que as distâncias entre duas séries temporais discretizadas são equivalentes às distâncias das mesmas séries no seu estado original. Essa prova matemática se refere ao cálculo da distância mínima

(*Lower Bounding*) entre uma série temporal discretizada ou outra no seu estado original. Além disso, o cálculo da distância entre duas subsequências é mais rápido na série discretizada. Por outro lado, não existe perda de informação no processo de redução de numerosidade, pois, a sequência dos símbolos omitidos pode ser facilmente recuperada por meio da manipulação correta dos índices da posição dos símbolos.

#### 4.1.2 Discretizadores

Esta seção apresenta os algoritmos de representação simbólica, SAX e SFA, utilizados para discretizar os dados dos sensores inerciais. Esses algoritmos são aplicados sobre as janelas de tempo definidas na etapa de segmentação. Geralmente, o formato da base de dados segue o padrão definido na Figura 4.3. Esse padrão mostra a diferença entre uma série temporal, sequência e subsequência. As subsequências são segmentos, de tamanhos iguais, que compõem uma sequência ou subsérie. Um conjunto de sequências, por sua vez, compõe uma série temporal. As subsequências são equivalentes às janelas de tempo explicadas na Seção 3.1. Sendo assim, no processo de discretização, a subsequência é representada por uma palavra e uma sequência é representada por um conjunto de palavras.



**Figura 4.3** Exemplo da diferença entre sequência e subsequência. Fonte [95].

##### 4.1.2.1 SAX - Symbolic Aggregate Approximation

O SAX [38] é um algoritmo de discretização simbólico na qual os símbolos são representados por palavras. Cada palavra é representada por um conjunto de letras pertencentes a um alfabeto de tamanho limitado. O SAX é composto por 4 etapas. A primeira trata da normalização da série temporal, onde os valores devem ser normalizados em torno da média 0 e desvio padrão 1, conforme mostra a fórmula da Equação 4.1.

$$x'_i = \frac{x_i - \mu}{\sigma} \quad \text{Equação 4.1}$$

onde  $x_i$  representa o valor da série temporal na posição  $i$ ,  $x'_i$  é o resultado da normalização,  $\mu$  é a média e  $\sigma$  é o desvio padrão dos dados da janela de tempo.

A segunda etapa trata da aplicação de um algoritmo de aproximação numérica, denominado PAA (*Piecewise Aggregate Approximation*) [197], com o propósito de reduzir a dimensionalidade dos dados. Nesse caso, as sequências são divididas em subseqüências  $M$  de igual tamanho, onde cada subseqüência é representada pela média de seus pontos. Essa aproximação torna o SAX o redutor natural de ruídos equivalente aos filtros de média móvel. A Equação 4.2 mostra a fórmula matemática do PAA aplicada em uma subseqüência.

$$\bar{x}_i = \frac{M}{n} \sum_{j=\frac{n}{M}(i-1)+1}^{(n/M)i} x_j \quad \text{Equação 4.2}$$

Onde  $M$  é a quantidade de valores de uma subseqüência,  $n$  é a quantidade de valores da seqüência,  $x_j$  é um determinado valor na posição  $j$  e  $\bar{x}_i$  é a média de determinada subseqüência. Cada subseqüência definida pelo PAA representa uma letra. Dessa forma, a quantidade de subseqüências contidas em uma seqüência representa a quantidade de letras de uma palavra.

$\beta_i$	$a$	2	3	4	5	6	7	8
$\beta_1$		0.00	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15
$\beta_2$			0.43	0.00	-0.25	-0.43	-0.57	-0.67
$\beta_3$				0.67	0.25	0.00	-0.18	-0.32
$\beta_4$					0.84	0.43	0.18	0.00
$\beta_5$						0.97	0.57	0.32
$\beta_6$							1.07	0.67
$\beta_7$								1.15

Figura 4.4 Tabela estatística utilizada na definição dos breakpoints. Fonte [48].

A terceira etapa trata da definição das letras de uma palavra com base em um alfabeto de tamanho limitado. Cada letra representa um intervalo de dados delimitado por *breakpoints*. Os *breakpoints* são definidos como sendo  $B = (\beta_1, \beta_2, \dots, \beta_n)$ , tal que  $\beta_{i-1} < \beta_i$ ,  $\beta_1 > -\infty$  e  $\beta_n < \infty$ . Os valores dos  $\beta_s$  são definidos por meio de uma tabela estatística que divide a curva gaussiana, derivada da normalização, em áreas iguais. Cada área é delimitada pelo intervalo  $[B_{j-1}, B_j)$ . Em outras palavras, os pontos de quebra formam intervalos nos quais o primeiro e o último ponto representam  $-\infty$  e  $+\infty$ . A Figura 4.4 mostra a tabela estatística utilizada na definição dos *breakpoints*.

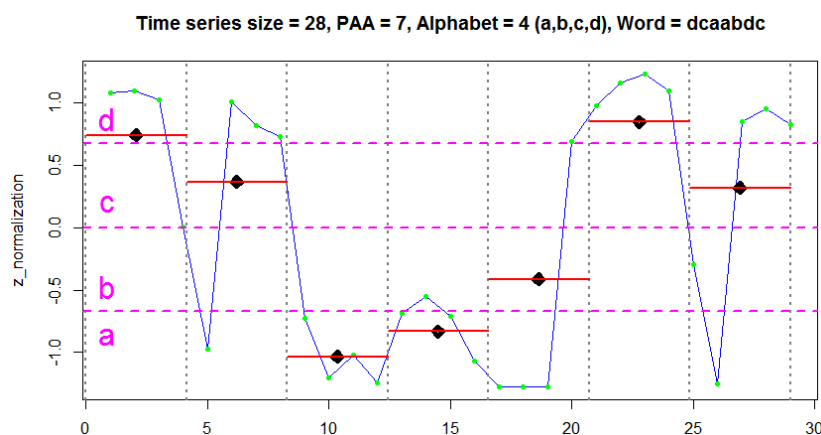
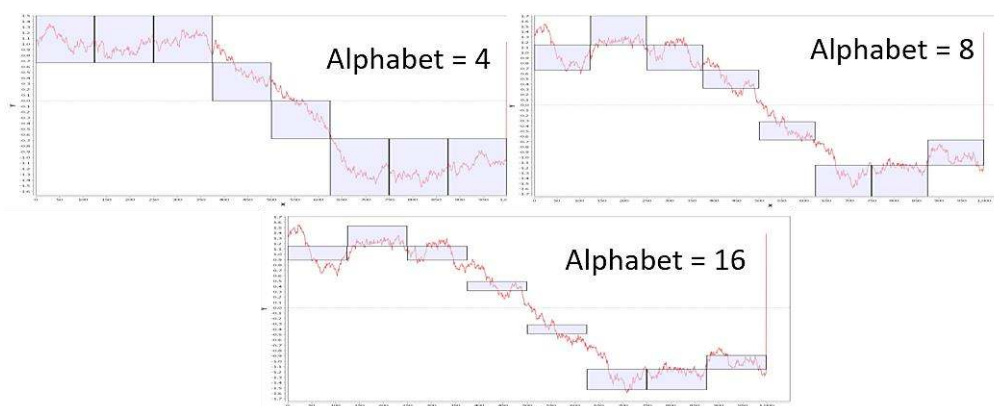


Figura 4.5 Exemplo de discretização de uma série temporal com o SAX. Fonte: autor.

Por fim, a última etapa é responsável pela formação das palavras, i.e., a concatenação das letras geradas na etapa anterior. A Figura 4.5 mostra um exemplo da aplicação do algoritmo SAX sobre uma subseqüência contendo 30 amostras (eixo x). Como pode ser observado, a normalização das amostras resultou numa distribuição Gaussiana com os valores da amplitude

variando num intervalo aberto entre -1.5 e 1.5. As amostras foram divididas em: 7 frações de tamanhos iguais pelo algoritmo PAA (linhas verticais) e em 4 intervalos pelos *breakpoints* (linhas horizontais). Desse modo, essa subsequência é representada por uma palavra com 7 letras formadas por um alfabeto de tamanho 4 (a, b, c, d). Neste caso, a palavra equivalente a essa subsequência é “dcaabdc”.

Shieh e Keogh [48] propuseram uma evolução do algoritmo SAX denominado de iSAX (*Indexable Symbolic Aggregate approxImation*). O iSAX surgiu com o propósito de eliminar a definição manual do parâmetro de entrada referente ao tamanho do alfabeto. A eliminação desse parâmetro permite que o iSAX seja capaz de representar uma subsequência por meio de várias palavras com diferentes resoluções, ou seja, palavras compostas por alfabetos de diferentes tamanhos. Isso significa que, quanto maior a resolução da palavra, maior é a aproximação da representação simbólica com a subsequência original. Para ilustrar, a Figura 4.6 mostra um exemplo de uma subsequência discretizada com o iSAX onde mostra diferentes resoluções representadas pela quantidade de letras das palavras.



**Figura 4.6** Exemplo visual de uma discretização de uma subsequência com 3 resoluções do algoritmo iSAX.  
Fonte: autor.

#### 4.1.2.2 SFA – Symbolic Fourier Approximation

O SFA [39] é um algoritmo de representação simbólica que também utiliza palavras compostas por um alfabeto de tamanho limitado. As principais diferenças entre o SFA e o SAX são: a) o SFA utiliza o algoritmo de aproximação numérica DFT em vez do PAA; b) a normalização dos dados no SFA é opcional; e c) o SFA possui uma técnica, denominada MCB (*Multiple Coefficient Binning*), para definir as posições dos *breakpoints*.

A Figura 4.7 mostra um exemplo visual da diferença entre o SAX e o SFA. Como pode ser observado, o SFA, visualmente, possui maior capacidade de aproximação da subsequência original. Este fato é comprovado por Schäfer e Höggqvist [39] por meio do cálculo da distância do limite inferior entre o SFA e o SAX/iSAX. Os resultados mostram que a similaridade entre duas palavras geradas pelo SFA possui menos falsos alarmes comparado à similaridade entre duas palavras geradas pelo SAX/iSAX. Isso é possível porque o DFT tem maior capacidade de aproximação (estatisticamente mais significativa) quando comparado com o PAA.

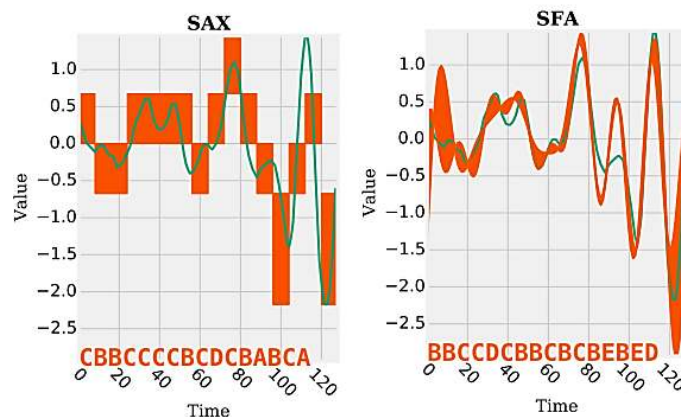


Figura 4.7 Exemplo visual da diferença entre o SAX e o SFA. Fonte [194].

O algoritmo SFA é dividido em duas fases: aproximação e quantização. Como mostrado na Figura 4.8, a fase de aproximação consiste na transformação da subsequência original em outra subsequência composta pelos coeficientes, reais e imaginários, extraídos da DFT (ver Seção 3.5.2). Essa fase também funciona como redução de ruído do sinal, pois os coeficientes extraídos do DFT equivalem ao resultado da aplicação do filtro de passa-baixa (*Lowpass*) [39]. Com o propósito de otimizar o cálculo dos coeficientes, o SFA utiliza uma versão modificada do DFT denominada de MFT (*Momentary Fourier Transform*). A principal diferença entre um e outro é que o MFT é capaz de calcular os coeficientes em tempo real, de forma recursiva, com base nos intervalos anteriores [194].

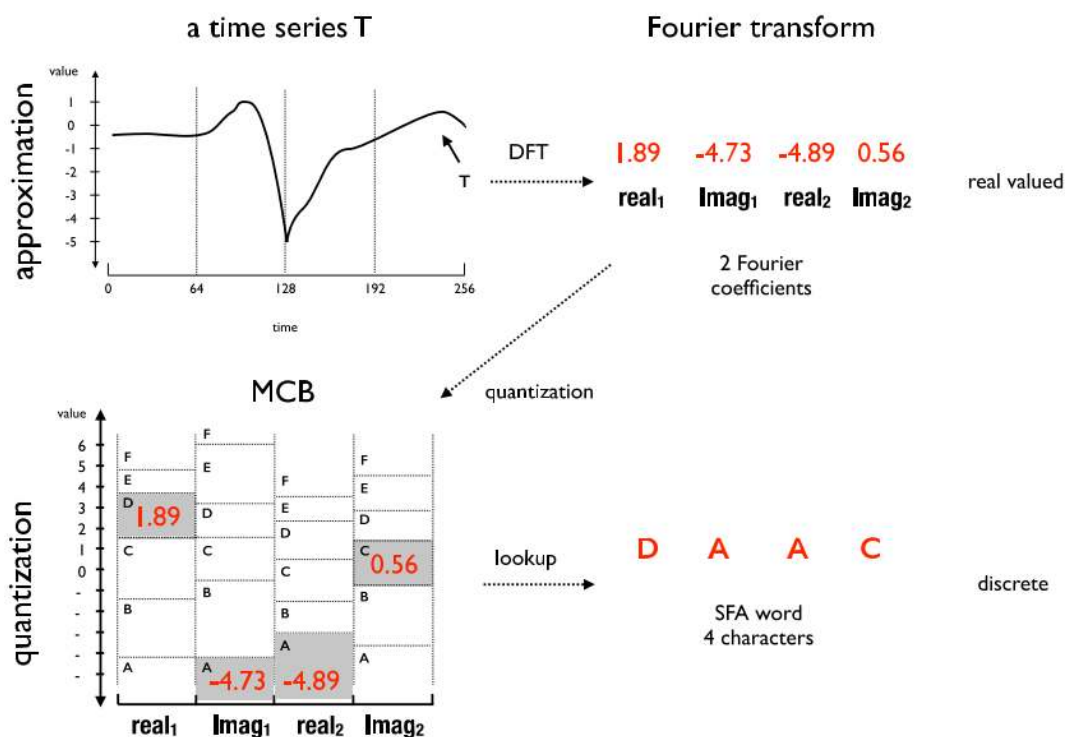


Figura 4.8 A subsequência T é aproximada utilizando o DFT e quantizada utilizando o MCB para, finalmente, gerar a palavra DAAC de tamanho 4 e alfabeto de tamanho 6. Fonte [194].

A fase de quantização consiste na aplicação da técnica MCB com o propósito de definir a melhor posição dos *breakpoints* que delimitam os intervalos das regiões de cada letra do alfabeto. Basicamente, as subsequências utilizadas para gerar o MCB são organizadas em forma

de matriz onde, a partir delas, são extraídos os coeficientes reais e imaginários pelo MFT. Sendo assim, cada coeficiente representa o limite de um intervalo de um *breakpoint*, ou seja, cada intervalo é formado por dois *breakpoints* que, por sua vez, representam um *bin*. Dentre os coeficientes gerados pelo MFT, nem todos são utilizados, pois o número de coeficientes depende dos tamanhos da palavra e do alfabeto passado como parâmetro para o SFA. Por exemplo, a Figura 4.9 mostra um exemplo visual de como o MCB é estruturado.

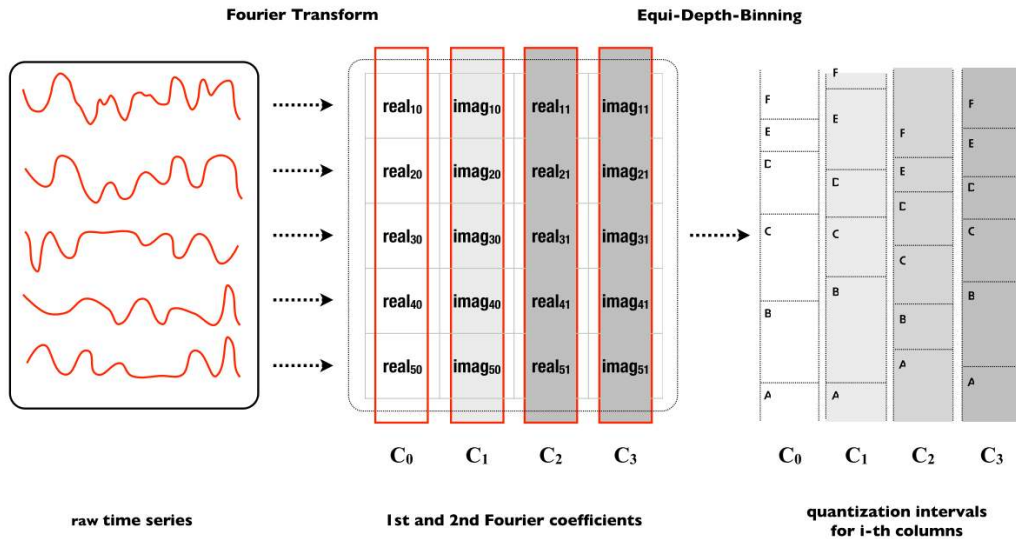


Figura 4.9 Exemplo de geração do MCB por meio dos coeficientes do MFT. Fonte [194].

Formalmente, o MCB é representado por uma matriz  $A = (a_{ij})_{i=1, \dots, N; j=1, \dots, l}$ , onde os valores de  $a_{ij}$  representam os primeiros coeficientes da transformada de Fourier sobre os valores das subsequências  $T_i$ , onde  $i \in [1, \dots, N]$ . Assim, a matriz  $A$  é representada da seguinte forma:

$$A = \begin{pmatrix} MFT(T_1) \\ MFT(T_i) \\ MFT(T_N) \end{pmatrix} = \begin{pmatrix} real_{11} & imag_{11} & \cdots & real_{12} & imag_{12} \\ real_{i1} & imag_{i1} & \cdots & real_{i2} & imag_{i2} \\ real_{N1} & imag_{N1} & \cdots & real_{N2} & imag_{N2} \end{pmatrix} = (C_1, \dots, C_N)$$

onde  $C_N$  representa a  $n$ -ésima coluna de coeficientes da matriz com valores de coeficientes reais e imaginários. Assim, cada coluna  $C_j \in A$ , onde  $j \in [0 \dots N]$ , representa um símbolo pertencente a um alfabeto finito  $\Sigma = \{symbol_1, \dots, symbol_c\}$  de tamanho  $c + 1$ . Cada símbolo do alfabeto  $\Sigma$  é definido pelos intervalos formados pelos coeficientes ordenados da coluna  $C_j$ . Os intervalos são representados por  $\beta_j(0) < \dots < \beta_j(c)$ , onde  $j \in [0 \dots N]$ ,  $\beta_i(0) = -\infty$  e  $\beta_i(c) = +\infty$ . Portanto, os *bins* do processo de quantização são definidos como:

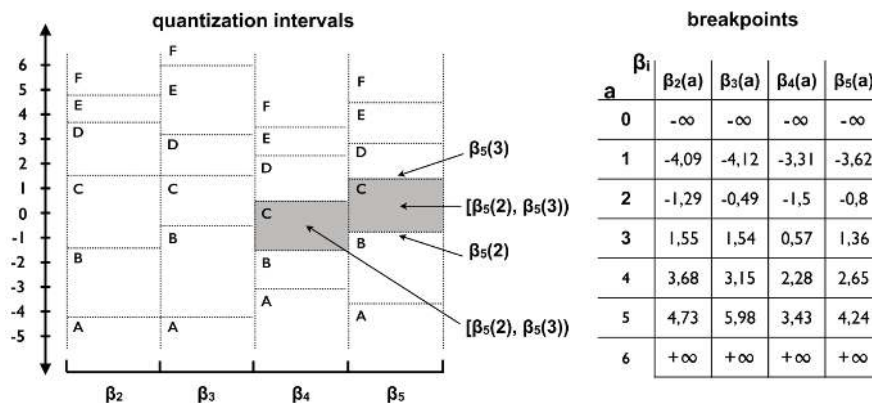
$$Bin_j(i) = [\beta(i-1), \beta(i)], \text{ para } j \in [0 \dots N], i \in [1 \dots C]$$

Cada *bin* é rotulado com o  $i$ -ésimo símbolo do alfabeto da seguinte forma:

$$Bin_j(i) \triangleq symbol_a \in \Sigma$$

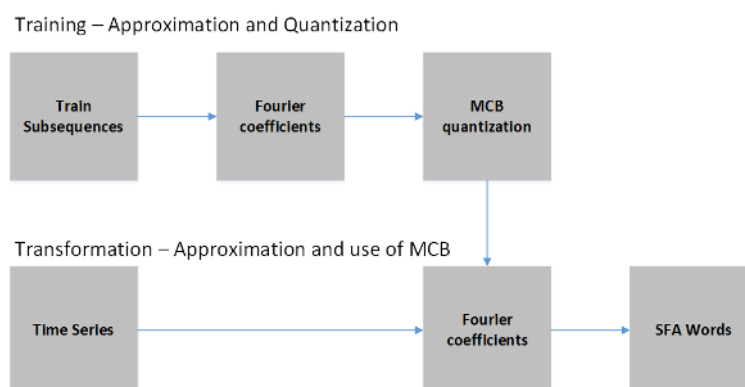
O exemplo da Figura 4.9 mostra a geração de um MCB a partir de cinco subsequências. Os parâmetros de entrada definidos foram: 4 para o tamanho da palavra e 6 para o tamanho do alfabeto. Portanto, para satisfazer o tamanho da palavra, os 4 primeiros coeficientes de todas as subsequências foram selecionados (colunas) e, para satisfazer o tamanho do alfabeto, os coeficientes de cada coluna foram ordenados em ordem crescente e, em seguida, as 7 primeiras

linhas  $(c + 1)$  de coeficientes foram selecionadas. A matriz de coeficientes resultante forma as seguintes *bins* descritos na Figura 4.10.



**Figura 4.10** Exemplo visual da distribuição dos bins e dos breakpoints em um MCB. Neste caso, o tamanho da palavra é 4 e do alfabeto é 6. Fonte [194].

Diferente do algoritmo SAX/iSAX, o SFA gera *bins* com intervalos diferentes para cada coluna de coeficientes. Esse comportamento reduz, consideravelmente, os riscos de perda de informação, pois essa estratégia consegue se adaptar melhor à variação dos dados. Além disso, o SFA sempre escolhe os primeiros coeficientes porque eles capturam os principais componentes da frequência, sendo que, os componentes restantes não afetam a precisão da aproximação significativamente [39]. A principal vantagem do uso desses coeficientes é a possibilidade de aumentar ou diminuir o nível de aproximação, apenas adicionando ou removendo os últimos coeficientes, respectivamente. Da mesma forma, esse efeito reflete nas palavras, pois adicionar ou remover uma letra também significa aumentar ou diminuir o nível de aproximação da subsequência. É importante ressaltar que os coeficientes são calculados apenas uma vez, não havendo necessidade de recalculá-los, caso necessite alterar o nível de aproximação da representação simbólica.



**Figura 4.11** Resumo de todas as etapas de processamento do SFA. Adaptado de [194].

A Figura 4.11 descreve o fluxo das etapas do SFA. Como pode ser observado, o MCB é criado com base em um conjunto de amostras de subsequências da série temporal. Isso significa que o MCB deve ser, inicialmente, treinado com subsequências suficientemente representativas. Após o treinamento, o MCB pode ser utilizado no processo de transformação e geração das palavras da série temporal.

Por fim, o algoritmo do SFA possui os seguintes parâmetros de entrada: a) tamanho da janela de tempo; b) normalização dos dados em torno da média e desvio padrão (verdadeiro ou falso); c) tamanho da palavra e d) tamanho do alfabeto.

## 4.2. Vantagens do Domínio Discreto no contexto HAR baseado em Smartphones

A principal vantagem das características do domínio discreto está na redução da dimensionalidade dos dados. Nos domínios de tempo e frequência, o processo de redução de dimensionalidade ocorre após a etapa de extração de características por meio da aplicação de métodos como CFS e KDA com o propósito de definir os melhores atributos. No caso do domínio discreto, este processo ocorre de forma natural na etapa de extração de características. Isso elimina a necessidade do uso posterior de métodos extras para reduzir a dimensão dos dados. Nessa perspectiva, o processo de redução da dimensionalidade dos dados tem como propósito representar, através de um símbolo, um segmento de uma série temporal da forma mais compacta e eficiente possível. Isso abre portas para que uma quantidade massiva de dados de séries temporais seja reduzida a uma quantidade razoável e representativa de símbolos. A naturalidade com que a dimensionalidade dos dados é reduzida evidencia outra vantagem do domínio discreto relacionada com uma maior escalabilidade das soluções [200]. Isso torna essas características uma solução competitiva para resolver o desafio de escalabilidade das soluções de HAR apontado na literatura.

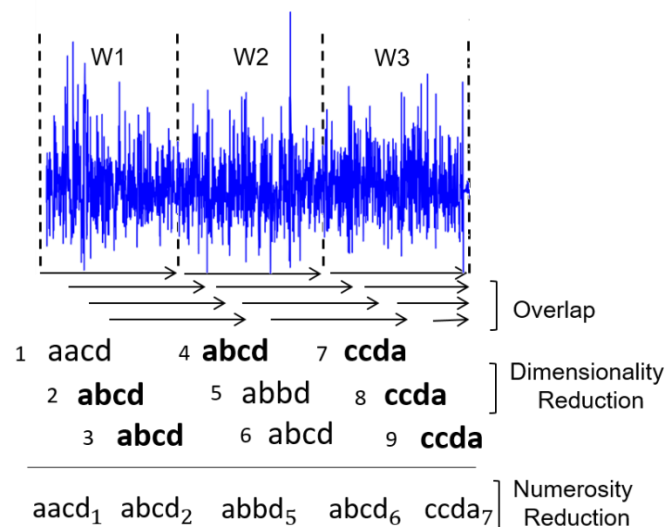


Figura 4.12 Exemplo de redução de dimensionalidade e numerosidade em uma série temporal discretizada e representadas por palavras. Neste caso, os números indicam os índices, as palavras em negrito indicam as palavras repetidas.

Durante a discretização das séries temporais, os símbolos são organizados de forma sequencial. Para cada um deles são atribuídos índices que representam as posições da série temporal original em que cada símbolo pertence. Essa organização sequencial de símbolos indexados permite reduzir, ainda mais, a quantidade de dados após o processo de discretização se, e somente se, houver subsequências com símbolos repetidos. A redução de símbolos indexados repetidos é conhecida como técnica de redução de numerosidade. Essa técnica agrega mais uma vantagem competitiva para as características do domínio discreto. A redução de numerosidade é possível devido os índices possuírem informações das posições de cada palavra



na série temporal original. Em outras palavras, dada uma subsequência de símbolos repetidos, apenas o primeiro símbolo deve ser mantido e os restantes podem ser suprimidos [201]. A Figura 4.12 mostra um exemplo de como ocorre a redução de dimensionalidade e numerosidade em uma série temporal discretizada.

O processamento dos dados no domínio discreto evidencia outra vantagem no que diz respeito à simplificação de algumas etapas da metodologia de HAR. Essa simplificação consiste na eliminação de algumas etapas de processamentos de dados, requeridas nos domínios de tempo e frequência, que são desnecessárias no domínio discreto. A primeira simplificação ocorre na eliminação do uso de métodos extras para reduzir a dimensionalidade dos dados, como foi abordado acima. A segunda simplificação consiste na eliminação da etapa de redução de ruídos dos sinais por meio do uso de filtros, visto que os algoritmos de representação simbólica são considerados redutores naturais de ruídos em séries temporais [39]. Por exemplo, a discretização realizada pelos algoritmos SAX [38] e SFA [39] equivalem, naturalmente, à aplicação dos filtros de média móvel [38] e *Lowpass* [39], respectivamente. Dessa forma, quanto menor o tamanho da palavra e do tamanho do alfabeto, maior é a redução dos ruídos do sinal. Obviamente, que a redução nos tamanhos dessas variáveis aumenta o risco da perda de informações. Por esse motivo, os parâmetros devem ser definidos de forma adequada para não comprometer a qualidade da aproximação dos algoritmos de representação simbólica.

Tais vantagens apresentadas acima diminuem consideravelmente a complexidade e o custo computacional das soluções de HAR. Além disso, essas vantagens abrem portas para utilização de novos algoritmos *online* e adaptativos para reconhecer atividades humanas no contexto de smartphones com recursos limitados de memória e processamento.

### 4.3. Algoritmos de Aprendizagem para o Domínio Discreto

O formato dos dados baseados nas características do domínio discreto tem direcionado novos estudos na perspectiva de adaptação e implementação de novos algoritmos capazes de atuar no contexto de classificação de séries temporais. Nesse sentido, os algoritmos de classificação baseados nessas características utilizam, em geral, a análise da frequência das palavras ao longo da série temporal. As distribuições de frequências das palavras podem ser obtidas por meio de um *bag of words* (BoW) ou histograma de palavras.

Um histograma pode ser definido como uma técnica de sumarização utilizada para organizar um conjunto de dados em termos da frequência de seus elementos. Desse modo, um histograma é formado por um conjunto de valores contidos em intervalos não-sobrepostos [35]. Visualmente, os histogramas são representados por um gráfico de barras onde o eixo x representa os elementos e o eixo y representa a frequência dos elementos. Os histogramas são comumente utilizados para capturar informações estatísticas de um conjunto de dados e, em seguida, utilizá-los em algum processo de tomada de decisão. A Figura 4.13 mostra um exemplo de histograma no contexto do domínio discreto.

No contexto de HAR, cada atividade possui um conjunto associado de palavras. A intuição é que a distribuição das palavras permita a discriminação das atividades humanas facilitando a atuação dos algoritmos de classificação. Nessa perspectiva, esta seção apresenta dois grupos de algoritmos, identificados na literatura, adaptados para atuar com as características do domínio discreto. O primeiro grupo trata dos algoritmos baseados em modelos

vetoriais como, por exemplo, SAX-VSM [40] e BOSS-VS [41] e, o segundo grupo trata dos algoritmos rasos convencionais como o BOSS-Model [42] e WEASEL [43].

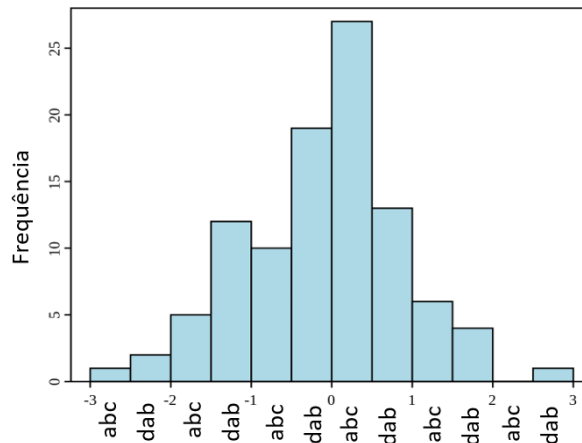


Figura 4.13 Exemplo de um histograma.

#### 4.3.1 Abordagens baseadas em Modelos Vetoriais

O modelo vetorial [202], comumente utilizado na área de recuperação de informação, tem sido alvo de várias adaptações para o contexto de HAR em diversos trabalhos da literatura [60] [40] [42] [41]. Essa abordagem tem sido bastante utilizada devido à facilidade de manipular textos. Em resumo, o modelo vetorial é utilizado para recuperar documentos relevantes de uma coleção com base em uma consulta composta por palavras chaves. No contexto de HAR, é possível fazer as seguintes associações e adaptações:

1. As atividades humanas podem ser traduzidas como a coleção de documentos.
2. As palavras geradas para cada atividade por meio do processo de discretização é equivalente as palavras do corpo de um documento.
3. Uma consulta pode ser traduzida como as palavras de uma série temporal não rotulada, onde objetivo é descobrir o nível de similaridade delas com as palavras de cada atividade.

Formalmente, o modelo vetorial é representado por vetores no espaço  $R^n$ , onde  $n$  representa a dimensão do espaço  $R$ . Cada dimensão é representada por um vetor de atividades  $A = (a_1, a_2, \dots, a_n)$ , onde  $a$  representa uma atividade  $n$  representa o número de atividades. Cada atividade  $a_n$  contém um vetor de palavras  $p = (p_1, p_2, \dots, p_j)$ , onde  $p$  representa uma palavra na posição  $j$ . Cada palavra  $p_j$  possui um peso  $w_{i,j}$  associado, onde  $w_{i,j} \geq 0$ . Além disso, cada atividade  $a_n$  será representada por um vetor de pesos  $W_j = (w_{1,j}, w_{2,j}, \dots, w_{i,j})$  onde  $i$  é a quantidade de todas as palavras distintas que aparecem na atividade  $a_n$  e  $j$  é a dimensão do vetor. O peso  $W_j$  é calculado levando em consideração a estratégia  $tf * idf$ , ou seja, a multiplicação da frequência das palavras,  $tf$ , pela sua importância,  $idf$ . A frequência da palavra  $tf_{p,A}$  é calculada de acordo com a Equação 4.3:

$$tf_{p,A} = \begin{cases} \log(1 + f_{p,A}), & \text{se } f_{p,A} > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (\text{Equação 4.3})$$

onde,  $f_{p,A}$  é a frequência da palavra na atividade  $A$  e,  $tf_{p,A}$ , é a frequência logarítmica da palavra (normalizada). A importância da palavra é calculada por meio do IDF (*Inverse Document Frequency*) que é dado pela Equação 4.4:

$$idf_{p,T} = \log \frac{T}{|\{A \in T: p \in A \wedge f_{p,A} > 0\}|} \quad (\text{Equação 4.4})$$

onde,  $idf_{p,T}$  é a importância da palavra  $p$  na série temporal  $T$  que representa os dados de todas as atividades disponíveis (coleção de documentos), para todo  $f_{p,A} > 0$ . Portanto, o  $tf * idf(p, A, T)$  da palavra  $p$  na atividade  $A$  é definido pela Equação 4.5:

$$tf * idf(p, A, T) = tf_{p,A} * idf_{p,T} = W_j \quad (\text{Equação 4.5})$$

A matriz de pesos  $W$  representa os  $tfidfs$  de todas as atividades, conforme mostrado na Tabela 4.1.

**Tabela 4.1 Exemplo da matriz de pesos gerados pelos TF-IDF.**

	<b>Atividade 1</b>	<b>Atividade 2</b>
<b>accbb</b>	0.023	0.0
<b>cdaaa</b>	0.14	0.0
...	...	...
<b>ddbca</b>	0.0	0.010

Para calcular o vetor de peso da consulta, o procedimento é o seguinte: calcula-se o  $idf$  das palavras da consulta normalmente, porém para o cálculo dos valores do  $idf$ , reutiliza os mesmos valores dos  $idfs$  das palavras utilizadas para calcular a matriz de peso.

Por fim, uma medida de similaridade é aplicada entre a matriz de pesos e o vetor da consulta com o propósito de descobrir a atividade mais próxima (ou similar) a consulta. Dessa forma, a similaridade entre dois vetores de pesos é dada pela seguinte fórmula do cosseno:

$$similaridade(Q, A) = \frac{\vec{Q} \cdot \vec{A}}{\|\vec{Q}\| \cdot \|\vec{A}\|} = \frac{\sum_{p \in Q} tf(p, Q) \cdot tfidf(p, A)}{\sqrt{\sum_{p \in Q} (tf(p, Q))^2} \sqrt{\sum_{p \in C} (tfidf(p, A))^2}}$$

Para dois vetores  $\vec{Q}$  (consulta) e  $\vec{A}$  (atividade), a medida de similaridade entre cossenos é baseada no somatório do produto interno entre esses vetores divididos pelas produto das normas do vetor da consulta e da atividade. Dessa forma, o rótulo da série mais próxima obtido pela métrica de similaridade é atribuído à série não-rotulada. As seções seguintes apresentam dois algoritmos baseados no modelo vetorial adaptado para séries temporais: o SAX-VSM [40] e o BOSS-VS [41].

#### 4.3.1.1 SAX-VSM - Symbolic Aggregate approxImation in Vector Space Model

O SAX-VSM [40] é um algoritmo de classificação de séries temporais baseado no SAX. Este algoritmo é dividido em duas fases: treinamento e classificação. A fase de treinamento segue as seguintes etapas:

1. Discretizar as séries temporais referentes a cada atividade com o algoritmo SAX. Como parâmetros devem ser escolhidos o tamanho da janela de tempo, PAA e alfabeto.
2. Identificar as palavras distintas e agrupá-las por atividade nos seus respectivos BoWs (*bag of words*) associadas suas respectivas frequências.
3. Calcular o vetor de peso de cada BoW baseado no  $tf * idf$ .
4. Montar a matriz de pesos com base nos vetores de pesos para cada atividade.

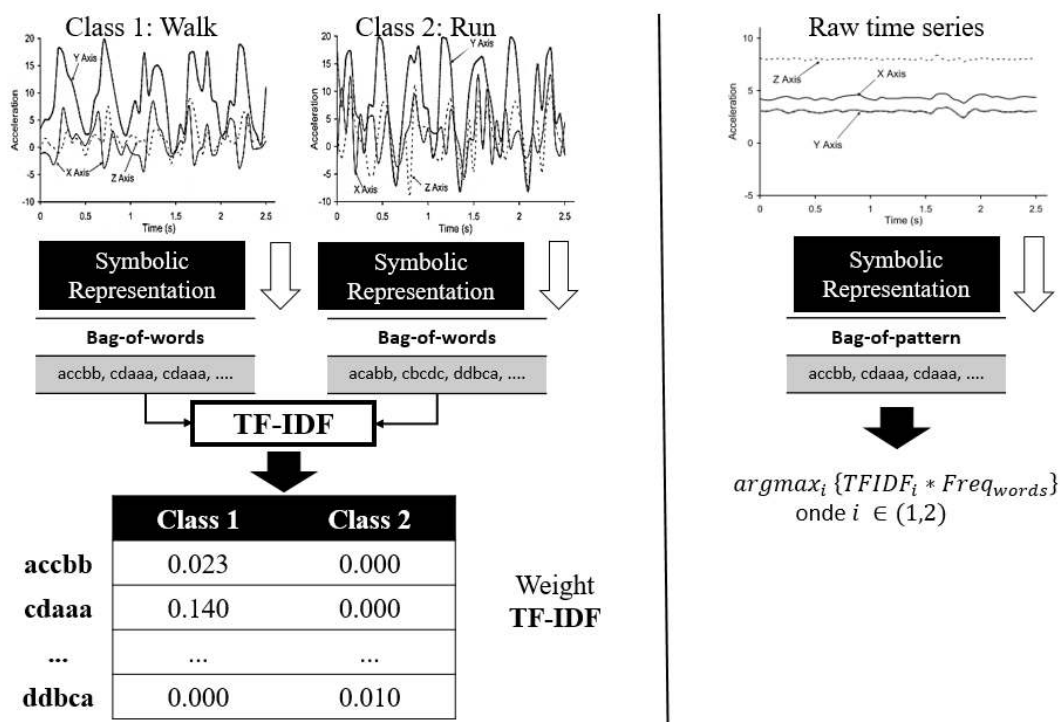


Figura 4.14 Exemplo das etapas de processamento de dados do algoritmo SAX-VSM. Fonte [40].

Em seguida, a fase de classificação segue as seguintes etapas:

1. Discretizar a série temporal referente à nova atividade com o SAX com os mesmos parâmetros definidos na etapa de treinamento.
2. Calcular o vetor de peso para as novas palavras da consulta geradas pelo SAX.
3. Comparar o vetor de peso da consulta com a matriz de peso gerada na etapa de treinamento utilizando a fórmula da similaridade dos cossenos. Classificar o vetor de pesos da consulta com base na maior similaridade com os vetores de pesos das atividades.

A Figura 4.14 mostra um exemplo das etapas de treinamento e classificação do SAX-VSM.

#### 4.3.1.2. BOSS-VS - Bag-Of-SFA-Symbols in Vector Space

O BOSS-VS [41] é um algoritmo de classificação de séries temporais baseado no SFA. Esse algoritmo também é dividido nas fases de treinamento e classificação semelhantes às etapas de treinamento e classificação do SAX-VSM descritas na Figura 4.14. A principal diferença entre um e outro é que o BOSS-VS usa SFA na fase de discretização e o *Bag of Patterns* na fase de representação de dados. A fase de treinamento segue as seguintes etapas:

1. Discretizar as séries temporais referentes a cada atividade com o algoritmo SFA. Como parâmetros devem ser escolhidos os tamanhos da janela de tempo, da palavra e do alfabeto. Também, deve-se informar se os dados devem ser normalizados ou não.
2. Criar os histogramas para cada atividade. No BOSS-VS esses histogramas são conhecidos como BOSS (*Bag-Of-SFA-Symbols*). O conjunto de todos histogramas é denominado de BOP (*Bag of Patterns*). A estrutura de geração do BOP pode ser vista na Figura 4.15. A Figura é dividida em quatro itens: (a) dados brutos da série



$$dist(BOSS_1, BOSS_2) = \sum_{k=1}^n (BOSS_1(f(p_k)) - BOSS_2(f(p_k)))^2_{BOSS_1(f(p_k)) > 0}$$

onde,  $BOSS_1$  e  $BOSS_2$  são os histogramas a serem comparados,  $f(p_k)$  é a frequência da palavra  $p_k$  e  $n$  é a quantidade de palavras no histograma. O cálculo da distância do BOSS não satisfaz a condição simétrica e nem a igualdade triangular. Isso significa que quando  $BOSS_1 \neq BOSS_2$ , a distância entre  $BOSS_1$  e  $BOSS_2$  é diferente da distância entre  $BOSS_2$  e  $BOSS_1$ . Além disso, o vizinho próximo também é afetado, pois quando  $BOSS_1$  é o mais próximo de  $BOSS_2$ , não significa que  $BOSS_2$  é o mais próximo de  $BOSS_1$ .

A Figura 4.16 mostra um exemplo de dois histogramas claramente diferentes um do outro. O resultado do cálculo da distância entre eles é  $dist(BOSS_1, BOSS_2) = 9$  e  $dist(BOSS_2, BOSS_1) = 14$ .

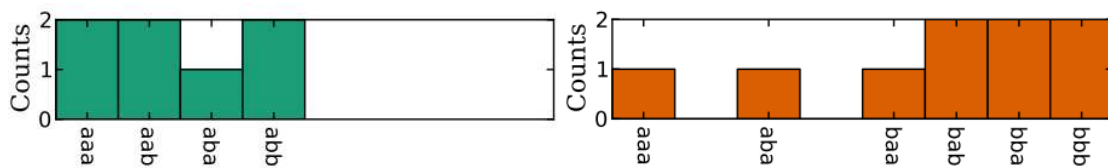


Figura 4.16 Exemplo de dois histogramas BOSS.

A etapa de treinamento no BOSS consiste na geração dos histogramas para cada atividade. A etapa de classificação consiste na análise de histogramas não rotulados com base no cálculo da distância BOSS utilizando o algoritmo 1-NN.

#### 4.4.2.2 Weasel - Word ExtrAction for time Series cLassification

O Weasel [43] é um algoritmo de classificação que utiliza uma estratégia linear baseada no algoritmo de regressão logística para classificar histogramas gerados a partir de uma série temporal. Basicamente, a regressão logística gera modelos com base no produto escalar entre os vetores de características com seus respectivos pesos. Tais vetores são extraídos dos histogramas de cada atividade. Em outras palavras, a regressão logística gera uma função linear que representa a distribuição das palavras de cada atividade e os pesos são utilizados para destacar as características relevantes e suprimir as irrelevantes.

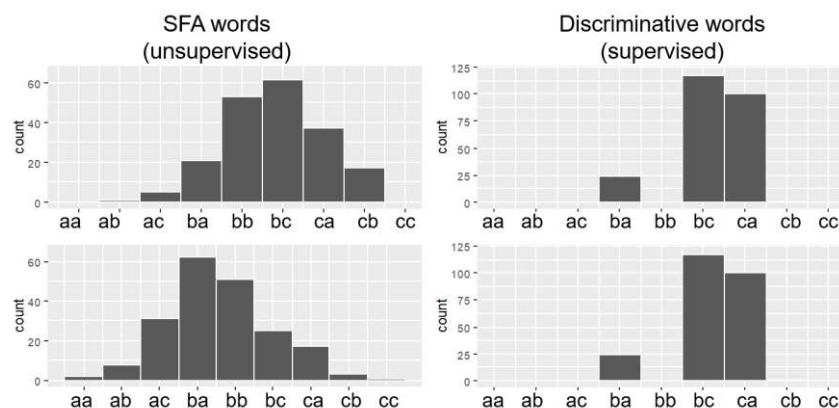


Figura 4.17 Diferença entre os histogramas gerados pelo SFA (esquerda) e SFA-W (direita). Fonte [43].

Algumas inovações foram propostas no Weasel na etapa de preparação dos dados antes de enviá-los para o método de regressão logística. A primeira inovação está relacionada com uma modificação no algoritmo SFA com o propósito de gerar palavras relevantes para

determinada classe. Isso significa que, além da função de discretização, o SFA-W (SFA modificado) também atua como um algoritmo de seleção de características. Nessa perspectiva, o SFA passa a atuar de forma supervisionada, sendo que, no momento da geração das palavras, o SFA-W verifica a relevância de cada uma delas em determinada classe. Em seguida, o SFA-W decide se a palavra gerada deve ser mantida ou descartada. A Figura 4.17 mostra um exemplo da diferença entre histogramas resultantes do SFA e do SFA-W.

O SFA-W modifica o SFA na etapa de aproximação e quantização. Na etapa de aproximação, a primeira exigência é que os dados devem ser obrigatoriamente normalizados em torno da média e desvio padrão. Diferentemente do SFA que seleciona os primeiros coeficientes sem olhar as classes, o SFA-W aplica o teste estatístico ANOVA F-test [203] para selecionar os coeficientes mais representativos do DFT para cada classe. O motivo da aplicação desse teste estatístico é que os melhores coeficientes de aproximação variam de dado para dado. O ANOVA F-test assume que os dados seguem uma distribuição normal, por isso a obrigatoriedade da normalização. Além disso, as subsequências utilizadas no treinamento do MCB são extraídas sem sobreposição com o propósito de garantir independência entre elas. Essa estratégia ajuda a diminuir o *overfitting* do modelo de classificação.

Na etapa de quantização, as colunas de coeficientes selecionadas pelo ANOVA F-test são divididas em duas partições. O objetivo é que cada partição concentre a maioria dos *bins*, dos quais, a maioria dos símbolos associados a cada um deles pertença à mesma classe. Para isso, o melhor ponto de corte entre uma partição e outra é definido pelo algoritmo de ganho de informação. O propósito deste algoritmo é minimizar a entropia entre as classes e maximizar o ganho de informação dos símbolos. Nesse ponto, as palavras são filtradas permanecendo apenas aquelas que são relevantes para determinada classe. A Figura 4.18 mostra um exemplo de como ocorre o processo de aproximação e quantização do SFA-W. Neste exemplo são utilizadas duas classes A e B. Os dados referentes a cada classe estão distribuídos nos coeficientes de Fourier extraídos da série temporal. Nesse caso, os coeficientes *real0* e *imag3* foram selecionados como os melhores pelo ANOVA F-test com base no melhor ponto de corte 0.46 definido pelo ganho de informação.

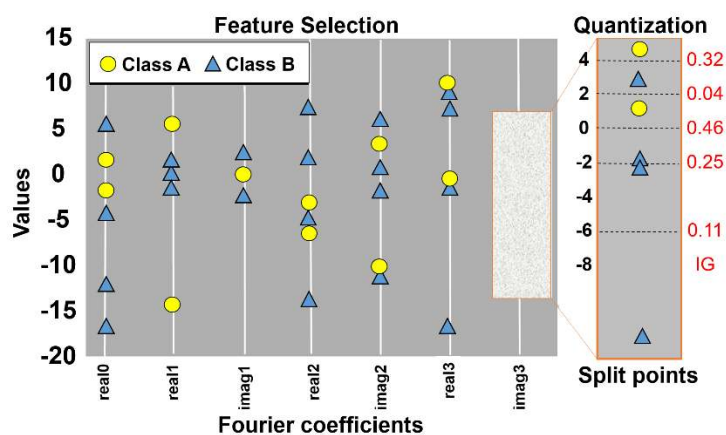


Figura 4.18 Exemplo de como funciona o processo de discretização do SFA-W. Fonte [43].

A segunda inovação do Weasel consiste na adição de novas informações úteis ao histograma BOP (*Baf-of-Pattern*). Tais informações consistem na adição de bigramas e novas palavras que representam tamanhos de janelas diferentes. Com relação ao bigrama, o Weasel introduziu o conceito *n*-gramas nas características de domínio discreto. Os bigramas são formados por duas palavras consecutivas, por exemplo, se o SFA-W gerar as palavras “*aab*”,

“*abc*”, “*acc*” e “*aaa*” então os bigramas para essas palavras são definidos como “*aab abc*” e “*acc aaa*”. Portanto, os bigramas e suas respectivas frequências também são incluídos nos histogramas BOP. A vantagem do bigrama é a adição de novos padrões impossíveis de serem capturados por uma única palavra. Por outro lado, a desvantagem dos bigramas é o aumento do espaço de características. Por isso, a escolha do algoritmo de regressão logística.

Na tentativa de diminuir o tamanho do espaço de características decorrente do uso de bigramas e de palavras de diferentes tamanhos por conta da variação de janelas de tempo, o Weasel aplica o teste estatístico *Pearson Chi-Squared* ( $\chi^2$ ). O propósito desse teste é selecionar as palavras relevantes dos histogramas de cada classe com base na análise das frequências das palavras (e bigramas). Este teste estatístico é aplicado sobre a distribuição de frequências das palavras de dois histogramas, de modo que se qualquer palavra da frequência observada for significativamente diferente da frequência esperada, então a palavra é eliminada do histograma. A decisão de eliminar uma palavra é baseada em um *threshold* definido pelo resultado do *Chi-Squared*. Após isso, os histogramas resultantes são utilizados para treinar o algoritmo de classificação regressão logística.

#### 4.4. Séries Temporais Multidimensionais

Os algoritmos de classificação mencionados acima são preparados para manipular apenas uma única série temporal. O problema é que os dados extraídos dos sensores inercias possuem múltiplas séries temporais, ou seja, as séries temporais referentes às coordenadas tridimensionais  $x$ ,  $y$  e  $z$ . Por essa razão, é necessário adotar estratégias capazes de adaptar esses algoritmos para atuar com múltiplas séries temporais ou múltiplas variáveis. Nesse sentido, esta tese de doutorado propõe duas estratégias.

A primeira estratégia consiste em uma alteração no formato da base de dados de forma que os algoritmos de classificação simbólicos (SAX-VSM, BOSS MODEL, BOSS VS e WEASEL) sejam capazes de empilhar as palavras de cada coordenada de forma sincronizada. A intuição dessa estratégia é que vários segmentos de diferentes séries temporais sincronizadas sejam organizados na mesma tupla com propósito de produzir o efeito de empilhamento. A vantagem dessa abordagem é que não é necessário fazer nenhuma alteração nas estratégias de *Bag of Patterns* utilizada pelo BOSS e nem da estratégia de *Bag of Words* utilizada pelo SAX-VSM. Por essa razão, qualquer um dos algoritmos descritos neste capítulo pode ser utilizado. O formato da entrada de dados consiste na seguinte organização matricial das coordenadas  $x$ ,  $y$  e  $z$ .

$$x_{1j}, \dots, x_{(i+k)j}, y_{1j}, \dots, y_{(i+k)j}, z_{1j}, \dots, z_{(i+k)j}$$

$$x_{2j}, \dots, x_{(i+k)j}, y_{2j}, \dots, y_{(i+k)j}, z_{2j}, \dots, z_{(i+k)j}$$

$$x_{ij}, \dots, x_{(i+k)j}, y_{ij}, \dots, y_{(i+k)j}, z_{ij}, \dots, z_{(i+k)j}$$

onde  $x_{ij}$ ,  $y_{ij}$ ,  $z_{ij}$  representa a série temporal de cada coordenada,  $i$  representa a posição dos valores da série,  $k$  representa tamanho da janela e  $j$  representa a posição da coluna. Em resumo, a matriz significa que  $x$ ,  $y$  e  $z$  são divididos em segmentos do mesmo tamanho para cada coordenadas. Depois, os segmentos são concatenados um com o outro. O efeito do empilhamento funciona em virtude dos algoritmos usando o tamanho de janela  $k$ , de modo que as palavras serão computadas independentemente para  $x$ ,  $y$  e  $z$ . Assim, essa estratégia evita



processamentos extras relacionadas a discretização isolada de cada série temporal para, posteriormente, as palavras serem empilhadas em um único histograma.

A segunda estratégia consiste no uso de técnicas de fusão de dados como a magnitude e o PCA (*Principal Component Analysis*) com o propósito de transformar as séries temporais multidimensionais em unidimensionais para utilizar os algoritmos de representação simbólica existentes na sua forma natural. No caso da magnitude, a fórmula utilizada para fundir as coordenadas é baseada na Equação 4.6:

$$M(X_i) = \sqrt{x^2_i + y^2_i + z^2_i} \quad (\text{Equação 4.6})$$

Onde,  $x$ ,  $y$  e  $z$  são as coordenadas dos sensores inerciais e  $X_i$  o conjunto de dados das séries temporais. No caso do PCA, apenas a primeira componente é considerada uma vez que ela contém as principais informações das componentes geradas.

Além das estratégias propostas acima, a literatura apresenta dois algoritmos derivados do WEASEL e do BOSS para manipular séries multidimensionais. O primeiro é o algoritmo Weasel+MUSE proposto por Schäfer e Leser [204]. O Weasel+MUSE apresenta um diferencial com a aplicação da técnica de derivação [205] de histogramas. Essa técnica consiste em relacionar os histogramas de cada coordenada por meio do cálculo da diferença entre os histogramas vizinhos de cada dimensão. O Weasel+MUSE utiliza como base a estrutura do Weasel, ou seja, todas as vantagens e desvantagens referentes ao uso de bigramas, tamanho de janelas e seleção de características são incorporados no Weasel+MUSE. O segundo é o algoritmo denominado *MBOSS* (*Multivariate BOSS*) proposto por Quispe [206]. O MBOSS é uma extensão do algoritmo BOSS. Nesse caso, as palavras geradas a partir de séries temporais diferentes são empilhadas em um único histograma. Nesse método, cada palavra possui um identificador que representa a coordenada a qual a palavra pertence. Nesse caso, o histograma resultante é uma fusão dos histogramas gerados para cada coordenada de forma separada. A Figura 4.19 mostra um exemplo do *MBOSS*.

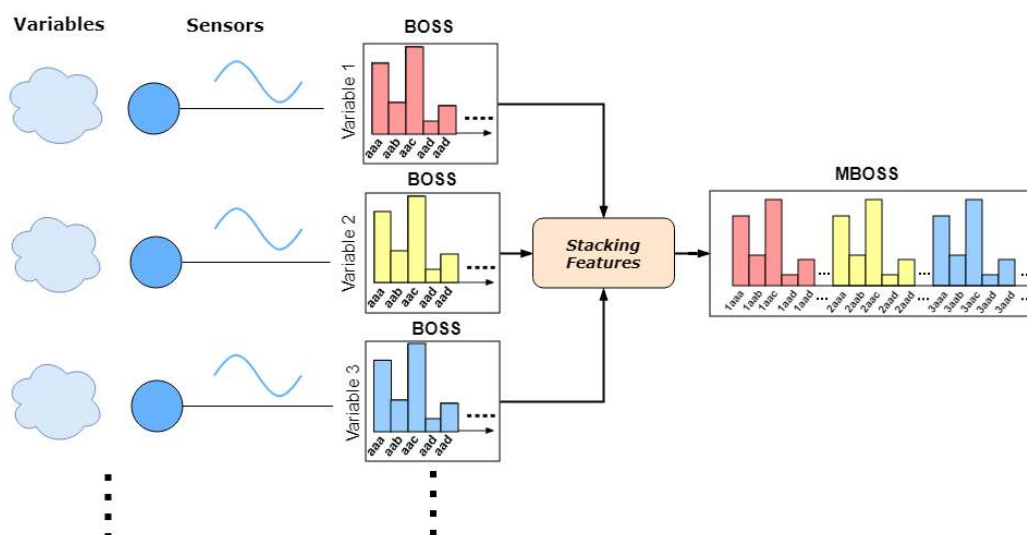


Figura 4.19 Empilhamentos das palavras de diferentes coordenadas.

No caso dos algoritmos de classificação rasos, baseados nas características do domínio do tempo e da frequência, a estratégia adotada para manipular múltiplas séries temporais é mais simples. As séries temporais são tratadas na etapa de extração de características por meio da

concatenação dos vetores de características de cada coordenada. No caso dos algoritmos baseados em redes neurais profundas, cada coordenada representa uma entrada diferente para rede neural e o processo de fusão é feito automaticamente nas camadas escondidas.

## 4.5. Trabalhos Relacionados

Os algoritmos de representação simbólica têm sido utilizados para reconhecer padrões recorrentes (*Motifs*) [38] ou anômalos (*Discords*) [207] [201] em séries temporais. No contexto de HAR são poucos os trabalhos que utilizam os algoritmos de representação simbólica [45] [44] [208]. Figo et al. [45], por exemplo, utilizam o algoritmo SAX para discretizar as séries temporais. O processo de classificação das atividades ocorre por meio do cálculo da distância entre os símbolos. Nesse caso, foram utilizadas as distâncias euclidiana, *Levenshtein* [209] e *Dynamic Time Warping* (DTW) [210]. Nos experimentos foi realizada uma análise comparativa entre as características do domínio discreto e 15 características do domínio do tempo (e.g. média, variância e desvio padrão) e 6 características do domínio da frequência (e.g. entropia e energia). Os dados utilizados foram obtidos do sensor acelerômetro e as atividades avaliadas foram pular, correr e caminhar.

Siirtola et al. [44] propuseram um novo método denominado de SAXS (*Symbolic Aggregate approXimation Similarity*). O SAXS consiste na definição de protótipos para cada atividade, sendo que cada um deles é representado por uma palavra genérica denominada de *template*. O processo de classificação das atividades ocorre por meio do cálculo da similaridade entre uma nova palavra não rotulada e os *templates* de cada atividade. Assim, a nova palavra é classificada com base no *template* mais próximo, i.e., com maior similaridade. Nos experimentos foi realizada uma análise comparativa entre as características extraídas com o SAXS e algumas características do domínio do tempo (média, desvio padrão, mediana, quartis, mínimo e máximo) e da frequência (soma dos coeficientes e *zero crossing*). Os dados utilizados foram obtidos e cinco bases de dados públicas e as atividades avaliadas foram as de movimento (e.g. andar), gestos e natação. Os classificadores avaliados foram a árvore de decisão, KNN e *Naive Bayes*. Os resultados mostram que o SAXS é capaz de aumentar a acurácia em 3% e em até 10% se combinado as características do domínio discreto com os demais domínios de características.

Terzi et al. [208] também utilizam o SAX para extrair as características do domínio discreto. Na classificação das atividades, a solução utiliza uma adaptação do k-NN para verificar a distância mínima entre as palavras rotuladas e as não-rotuladas. Por fim, uma extensão do SAX é proposta com o propósito de manipular séries temporais multidimensionais. Basicamente, o que ocorre é uma combinação das distâncias mínimas de múltiplas coordenadas por meio do produto entre elas. Os experimentos foram realizados na base de dados UCI HAR e as atividades avaliadas andar, subir e descer escadas. Os resultados mostram acurácias de 99,54% e 97,07% para as respectivas classes de atividades.

## 4.6. Considerações Finais

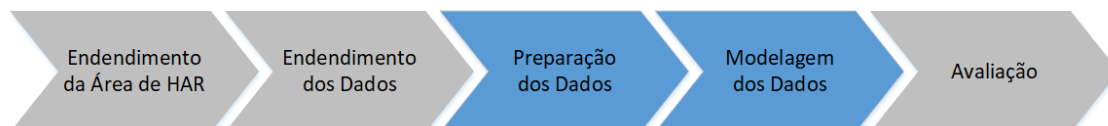
Este capítulo descreve os conceitos juntamente com as vantagens e as desvantagens do uso de dados discretos no contexto de HAR para smartphones. Além disso, este capítulo apresentou novos algoritmos de classificação adaptados para manipular dados simbólicos. Em seguida, nós apresentamos uma proposta alternativa para que os algoritmos de representação simbólica

tradicionais (que manipulam apenas séries temporais unidimensionais) sejam capazes de lidar com as séries temporais multidimensionais. Por fim, nós apresentamos os poucos trabalhos encontrados na literatura que utilizam as características do domínio discreto.

Note que os algoritmos de classificação simbólicos apresentados neste capítulo ainda não foram utilizados na literatura para resolverem os problemas de HAR. Por esta razão, esta tese de doutorado mostra um novo caminho que pode ser explorado pela comunidade de HAR. Esse novo caminho é parte das contribuições discutidas na Seção 1.4 as quais geraram dois periódicos de qualis A1 na revista *Sensors* [49] [51], 1 capítulo de livro [52] e 1 artigo publicado em anais de conferência [51].

# CAPÍTULO 5

## APRENDIZAGEM ONLINE



Este capítulo apresenta os fundamentos relacionados à aprendizagem *online*. O capítulo inicia com uma visão geral das principais diferenças entre a aprendizagem *offline* e a *online*, incluindo a apresentação das vantagens do emprego dessa abordagem no contexto de HAR para smartphones. Em seguida, o capítulo mostra uma visão geral dos algoritmos *online* por meio da apresentação de conceitos. Depois, uma nova tendência desses algoritmos baseada na detecção de novidades ao longo do tempo é apresentada. Em seguida, são apresentados os trabalhos relacionados na perspectiva dos smartphones com sensores inercias. Por fim, o capítulo aborda as estratégias de avaliação dos algoritmos *online*.

### 5.1. Online HAR

As soluções de HAR vistas até agora são baseadas em algoritmos de classificação *offline*, ou seja, algoritmos que produzem modelos de classificação baseados em históricos de dados limitados por uma pequena amostra de atividades coletadas de um grupo específico de usuários. A limitação do histórico de dados dificulta a generalização dos modelos de classificação devido às constantes mudanças das atividades humanas ao longo do tempo devido, por exemplo, às variações no comportamento humano e nos ambientes físicos. Além disso, quanto maior o tamanho do histórico de dados, maior é o custo de processamento dos algoritmos nos smartphones. Na tentativa de resolver este problema, algumas soluções de HAR [30] [31] [32] recorrem a computadores com maior poder de processamento (e.g. servidores) para executar o treinamento dos modelos de classificação. No entanto, o envio de dados para os servidores via rede afeta negativamente o consumo de energia das baterias dos smartphones, tornando essa abordagem inviável. Desse modo, as desvantagens das soluções *offline* podem ser resumidas em:

1. Modelos de classificação estáticos limitados à distribuição da base de dados de treino;
2. A rotulação dos dados é uma tarefa que exige bastante esforço, sobretudo porque o cenário em que a área de HAR está inserida é bastante diversificado, o que torna o trabalho manual de um especialista humano impossível [211];
3. A transmissão dos dados para um servidor é dispendiosa na perspectiva do consumo de energia das baterias dos smartphones;

As desvantagens das soluções *offline* têm motivado o desenvolvimento de novas soluções direcionadas para o uso de algoritmos de aprendizagem de máquina *online* [34] [25] [26]. A estratégia utilizada pelos algoritmos *online* permite que os modelos de classificação sejam atualizados dinamicamente à medida que novos dados são coletados. Essa estratégia é baseada na extração e armazenamento somente de informações fundamentais a partir de um conjunto de dados (e.g. medidas estatísticas), de forma que os dados brutos processados não

precisam ser armazenados na memória. Essa estratégia elimina completamente a necessidade de manter um histórico de dados armazenado. Dessa forma, as vantagens dos algoritmos *online* em relação aos *offline* podem ser resumidas em:

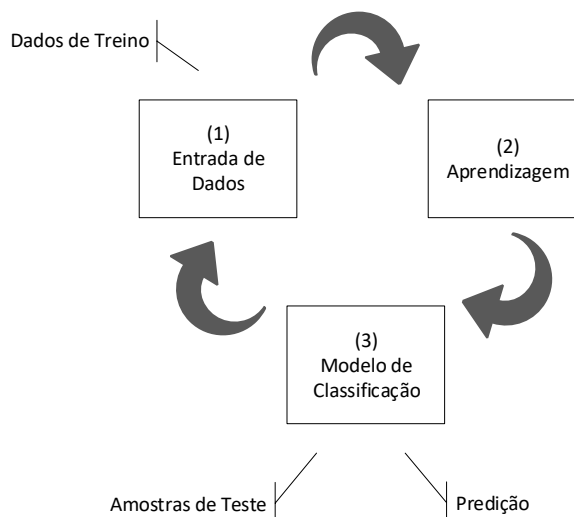
1. Processamento de dados contínuos, ilimitados e sequenciais que vão sendo produzidos ao longo do tempo, sendo impossível armazená-los completamente na memória;
2. Os algoritmos *online* atuam com pouca quantidade de memória sem a necessidade de manter armazenado um histórico de dados;
3. À medida que novos dados são processados, os algoritmos *online* extraem as informações fundamentais, descarta os dados brutos e toma decisões em tempo real;
4. Os algoritmos *online* não são mais limitados à distribuição da base de dados de treino, visto que os dados mudam ao longo do tempo e dados antigos podem tornar-se irrelevantes;
5. Os modelos de classificação são atualizados antes de sofrerem degradações consideráveis relacionadas com o erro de classificação;

Poucas soluções de HAR têm utilizado algoritmos *online* no contexto de smartphones equipados com sensores inerciais. Em vez disso, a literatura tem empregado o termo *online* para as soluções que foram totalmente implementadas no *smartphone*. Nessa perspectiva, as soluções são caracterizadas pelo uso de técnicas de baixo custo computacional (*lightweight*) com o propósito de viabilizar o processamento dos dados nos smartphones. Por exemplo, existem soluções que utilizam as características do domínio do tempo combinadas com algoritmos de classificação *offline* de baixo custo como Árvore de Decisão e *Naive Bayes* [78] [63] [83] [84]. Shoaib et al. [101] apresenta uma lista de trabalhos que seguem essa direção. Mesmo com essas tentativas, as desvantagens dos algoritmos *offline* continuam sendo mantidas. As próximas seções mostram como a aprendizagem *online* pode ser aplicada na área de HAR baseada em smartphones equipados com sensores inerciais.

## 5.2. Conceitos de Aprendizagem Online

A aprendizagem *online* é composta por algoritmos adaptativos capazes processar fluxo contínuo de dados (do inglês, *data streaming*) em diferentes velocidades, escalas e granularidades [35]. As etapas básicas desses algoritmos consistem na extração de informações fundamentais dos dados de forma a encontrar padrões ao longo do tempo que podem ser representadas por um modelo adaptativo. De modo geral, a estratégia de aprendizagem dos algoritmos *online* pode ser resumida no exemplo da Figura 5.1. Como pode ser observado, o algoritmo recebe como entrada as amostras de dados desconhecidas ao longo do tempo (bloco 1). Em seguida, o algoritmo analisa a amostra e atualiza o modelo de classificação (bloco 2). Por fim, o algoritmo realiza inferência sobre a nova amostra com base no modelo de classificação existente. Em seguida, o processo se repete conforme novas amostras vão sendo coletadas.

Formalmente, os algoritmos de *data streaming* podem ser definidos como [35]: um sequência de dados  $T_t = \{\vec{x}_t, y_t : y = f(\vec{x})\}$  onde  $t \in \{1, 2, 3, \dots, i\}$ ,  $\vec{x}_t$  representa o vetor de dados,  $y_t$  representa o rótulo que é predito pela função  $f(\vec{x})$ . A aprendizagem adaptativa é aplicada sobre uma sequência de exemplos  $\{\dots, E_{j-1}, E_j, \dots\}$  que produz uma sequência de hipóteses  $\{\dots, H_{j-1}, H_j, \dots\}$  onde cada hipótese  $H_j$  depende da hipótese anterior  $H_{j-1}$  e do exemplo  $E_j$ .



**Figura 5.1 Visão Geral dos Algoritmos Online. Adaptado de [36].**

Um olhar atento à literatura permite identificar dois grupos de algoritmos *online* [212] [36]. O primeiro grupo trata dos algoritmos de aprendizagem estacionária e o segundo grupo trata dos algoritmos de aprendizagem baseada na evolução dos dados.

Os algoritmos baseados na aprendizagem estacionária são caracterizados por atualizarem os modelos de classificação sempre que uma nova instância de dados ou um conjunto limitado de novas instâncias de dados estão disponíveis. Essa estratégia de aprendizagem é conhecida como aprendizagem “cega”, ou seja, não existem critérios que limitam o número de vezes em que os modelos de classificação sejam atualizados. Por outro lado, existem os algoritmos *online* que atualizam os modelos de classificação somente quando necessário. Essa estratégia de aprendizagem é baseada na observação da evolução dos dados ao longo do tempo por meio de técnicas capazes de detectar mudanças na distribuição dos dados.

No contexto dos sensores inerciais, uma mudança nos dados pode ser caracterizada pela variação estatística do sinal causada pelos diferentes padrões de atividades. Por exemplo, a atividade andar possui uma média e variância padrão. Quando essas informações mudam é provável que tenha ocorrido mudança de atividade ou o sinal do sensor tenha sofrido algum ruído. Portanto, a intuição por trás da detecção de mudança nos dados é, pontualmente, evitar possíveis erros de classificação quando os dados mudam e, também, evitar atualizações desnecessárias dos modelos de classificação. Essa estratégia é capaz de melhorar, ainda mais, a agilidade de resposta dos algoritmos *online*, tornando-os mais rápidos, inteligentes e “leves”, quando comparados aos algoritmos *online* da aprendizagem estacionária. Portanto, as técnicas de detecção de mudanças nos dados são fundamentais para as tomadas de decisões dos algoritmos *online* [212].

A literatura apresenta dois grupos de técnicas de detecção de mudanças compatíveis com séries temporais. O primeiro trata das técnicas que analisam amostra por amostra, como o CUSUM (*Cumulative Sum*) [212], SPC (*Statistical Process Control*) [212] e o Page Hinkley [212]. O segundo trata das técnicas que analisam um conjunto de amostras contidas em uma janela de tempo, como o ADWIN (*ADaptive sliding WINdow*) [35]. O algoritmo ADWIN, por exemplo, é amplamente utilizado pelos algoritmos *online* adaptativos (ver Seção 5.4.2) para detectar as mudanças dos dados.

### 5.3. Aprendizagem Online baseada em Detecção de Novidades

Os algoritmos discutidos na seção anterior são capazes, naturalmente, de detectar novos padrões (ou novidades) nos dados ao longo do tempo. O problema é que esses algoritmos não tratam as novidades de forma a dar significado para cada uma delas. Por exemplo, no contexto de HAR baseado em smartphones, as novidades podem representar uma nova atividade. No entanto, se essas novidades não forem “rotuladas”, uma possível decisão de um determinado algoritmo de recomendação, por exemplo, pode não fazer sentido. Para resolver esse problema é necessário que os algoritmos *online* recebam *feedbacks* externos que atribuam significado para o novo padrão descoberto. Nesse sentido, esse problema tem sido tratado com o uso de aprendizagem ativa a qual é baseada em *feedbacks* dados pelos usuários.

Os algoritmos *online* baseados em detecção de novidades são evoluções dos algoritmos *online* baseados na detecção de mudanças nos dados. A diferença entre eles está na adição de mais uma tarefa relacionada com a aprendizagem ativa e a adição de mais um componente relacionado com estimação de perdas do algoritmo. Gama et al. [212] apresentam uma arquitetura completa que engloba todos os componentes e tarefas de um algoritmo *online*. Existem quatro componentes principais incluindo memória, aprendizagem, detecção de mudanças e estimador de perdas e, também, das tarefas relacionadas com a entrada de dados, *feedback* do usuário, alarme da detecção de mudança e predição das amostras.

O primeiro componente é responsável pelo gerenciamento de memória, visto que os algoritmos *online* analisam somente dados recentes e descartam os dados antigos. Os dados recentes são armazenados temporariamente em um tipo de memória curta (*Buffer*). Os dados no *buffer* podem ser subdivididos em vários pedaços iguais (*chunks*) que, no contexto de séries temporais, podem ser definidos pelas janelas de tempo. O modelo de classificação é o único elemento que é mantido permanente na memória, porém, ele sempre sofre alterações ao longo do tempo.

O segundo componente está relacionado com a evolução dados sob quatro perspectivas. A primeira consiste na detecção de mudanças de conceitos nos dados ao longo do tempo (do inglês, *concept drift*) [26] que, em resumo, representa uma mudança na distribuição de probabilidades das informações fundamentais referentes aos dados um rótulo. A segunda consiste no surgimento de novos conceitos que podem representar novos rótulos. A terceira consiste no esquecimento dos rótulos que não são mais utilizados pelo modelo de classificação durante certo período de tempo. A quarta consiste na detecção de ruídos e anomalias que representam mudanças incoerentes nos dados. Os algoritmos *online* devem ser capazes de fazer uma distinção entre ruídos, anomalias e mudanças legítimas nos dados com o propósito de minimizar os falsos positivos durante a classificação das amostras. A Figura 5.2 mostra exemplos dos quatros conceitos acima mencionados. Quando uma mudança nos dados é detectada, um alarme é acionado e a fase de análise de dados é iniciada (aprendizagem).

O terceiro componente consiste no processo de aprendizagem contínua e dinâmica dos modelos de classificação. Esse processo é totalmente dependente das mudanças nos dados e está diretamente relacionado com os alarmes emitidos por essas mudanças. O processo de aprendizagem é composto por duas operações básicas: incremento e decremento. O incremento consiste na adição ou atualização de informações no modelo de decisão e o decremento consiste no esquecimento dessas informações. Tais operações são realizadas somente em partes específicas do modelo sem afetar as demais partes. Por exemplo, se a decisão a ser tomada for o

esquecimento de um rótulo, somente as informações fundamentais referentes ao rótulo selecionado serão excluídas, o restante do modelo permanece intacto.

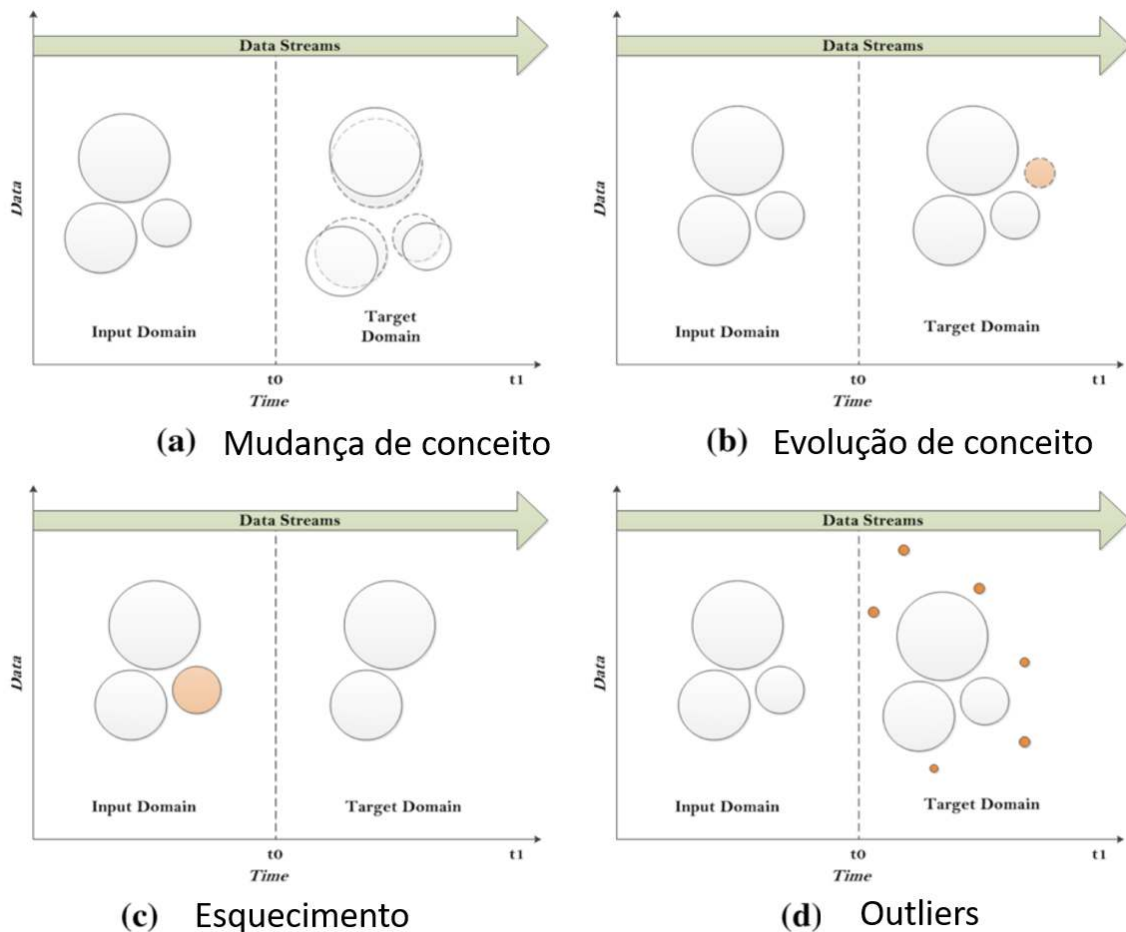


Figura 5.2 Mudanças na distribuição dos dados em data streaming. Adaptado de [26].

O quarto componente consiste no monitoramento da perda de precisão dos modelos de classificação por meio de funções de estimação de perda. Tais funções também podem auxiliar no ajuste de parâmetros do algoritmo de forma geral com o propósito de melhorar a precisão do modelo de classificação.

De modo resumido, os algoritmos de aprendizagem *online* baseados em detecção de novidades seguem três caminhos associados com a tomada de decisão dos modelos de classificação. Tais caminhos consistem em prever, diagnosticar e atualizar os modelos. Por exemplo, dada uma entrada de dados, se as informações fundamentais desses dados forem conhecidas, o algoritmo segue o caminho da predição. Senão, os dados são diagnosticados e, se houver mudanças de conceitos, o modelo de classificação é atualizado. A principal vantagem das técnicas de *data streaming* é a geração de modelos especialistas e adaptativos. No contexto de HAR, essa vantagem ajuda na resolução do problema relacionado com a diversidade de ambientes e atividades dos usuários discutidas na Seção 2.8.

## 5.4. Baselines

Esta seção apresenta detalhes dos algoritmos de classificação *online* utilizados como *baselines* para avaliação do algoritmo proposto por este trabalho, o NOHAR (ver Capítulo 6).



Basicamente, os algoritmos selecionados são baseados nas estratégias de aprendizagem estacionária, evolução dos dados e detecção de novidades.

#### 5.4.1 Aprendizagem Estacionária

No contexto da aprendizagem estacionária, a literatura apresenta o algoritmo *Hoeffding Tree* [213] como o algoritmo estado da arte na área de aprendizagem *online* [36]. O *Hoeffding Tree* é um algoritmo baseado em implementações otimizadas dos algoritmos de árvores de decisão *offline*, mais especificamente da VFDT (*Very Fast Decision learner*). O diferencial do *Hoeffding Tree* é a capacidade de induzir uma árvore de decisão de forma incremental usando uma estratégia denominada de *Hoeffding Bounds*. Desse modo, o algoritmo é capaz de analisar uma quantidade limitada de dados e extrair apenas as informações necessárias para gerar a árvore de decisão. Nesse sentido, somente a estrutura de dados da árvore de decisão é armazenada na memória e os dados brutos são completamente descartados após o processo de análise.

De modo geral, árvore de decisão é uma estrutura de dados capaz de analisar uma informação por diferentes caminhos dependendo dos valores particulares de cada atributo ou características. Os caminhos são divididos por meio de três tipos de nós: nó raiz, nós intermediários e nós folhas. Cada nó representa uma encruzilhada que possui dois ou mais caminhos que levam a outros nós. Todos os caminhos iniciam no nó raiz e terminam no nó folha.

O maior desafio de uma árvore de decisão é a definição de quando e como um nó deve ser dividido para formar novos caminhos. Nesse caso, a literatura utiliza estratégias baseadas na pureza dos atributos, ou seja, o ideal é utilizar somente atributos capazes de discriminar bem as classes do problema. Nessa perspectiva, a pureza de um atributo é medida usando a entropia da distribuição dos valores de cada atributo nas classes. Dessa forma, uma classe é representada por  $p = (p_1, p_2, \dots, p_n)$ , onde  $p_n$  representa a frequência do  $n$ -ésimo valor de um atributo na classe  $p$ . Assim, a equação 5.1 mostra o cálculo da entropia para uma classe:

$$\text{entropy}(p_1, p_2, \dots, p_n) = \sum_{i=1}^n -p_i \log p_i \quad \text{Equação 5.1}$$

onde  $\text{entropy}(p_1, p_2, \dots, p_n)$  representa o somatório das entropias das frequências dos valores de cada atributo na classe  $p$ . O ganho de informação para cada classe é medido por meio da subtração entre a entropia da base de dados e a entropia de cada classe resultante da equação 5.1. A entropia da base de dados é medida com base na frequência dos valores  $p_1, p_2, \dots, p_n$  das classes disponíveis na base de dados.

A construção de uma árvore de decisão utilizando a aprendizagem em lote (*batch*) por meio de um algoritmo *offline* é mais simples porque o ganho de informação de cada atributo é baseado na distribuição dos dados da base de dados de treino. Por outro lado, na perspectiva da aprendizagem de *data streaming* por meio de algoritmos *online*, a construção da árvore de decisão é um desafio, visto que a quantidade de dados é infinita e a distribuição dos dados pode mudar ao longo do tempo. Por este motivo, Hulten and Domingos [213] propuseram a estratégia *Hoeffding Bounds* descrita na equação 5.2:

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}} \quad \text{Equação 5.2}$$

onde  $\epsilon$  representa uma estimativa do ganho de informação real dos dados. O valor de  $R$  representa uma faixa de valores que o ganho de informação pode alcançar (valor fixo). O valor

de  $\delta$  representa a probabilidade do atributo correto ser escolhido a cada nó da árvore. O valor de  $\delta$  varia ao longo do tempo. Por fim, o valor de  $n$  representa o número de amostras.

A principal função de  $\epsilon$  consiste em decidir se um nó deve ser dividido ou não. Essa decisão é tomada quando a diferença dos valores do ganho de informação entre o melhor atributo e o segundo melhor atributo é maior que  $\epsilon$ . À medida que o número de amostras cresce, o valor de  $\epsilon$  decresce na esperança de que o ganho de informação se aproxime o máximo do valor real. Esse é o principal princípio de indução do algoritmo *Hoeffding Tree*, mais detalhes sobre o algoritmo podem ser encontrado em [36].

O cálculo do ganho de informação para todas as amostras dos dados é computacionalmente custoso. Por este motivo, o *Hoeffding Tree* possui um importante parâmetro, denominado *Grace Period*, que define de quanto em quanto tempo (ou amostras) o ganho de informação deve ser recalculado. Hulten e Domingos [213] definem um valor padrão de 200 amostras. No contexto de HAR, o *Grace Period* é definido com base na quantidade de amostras de uma janela de tempo. Essa quantidade varia de acordo com a taxa de frequência da coleta de dados. Por exemplo, se os dados forem coletados a uma taxa de 20 Hz em uma janela de 5 segundos, o valor do *Grace Period* deve ser no mínimo 100 devido este valor ser o tamanho da janela de tempo.

Além disso, várias árvores *Hoeffding Tree* podem ser combinadas por um método *ensemble* utilizando as estratégias *Bagging* e *Boosting* discutidas na Seção 4.1. Ambas as estratégias foram adaptadas para o contexto de *data streaming* por Oza e Russell [214]. Além disso, Kohavi e Kunz [215] propuseram as árvores de opções (do inglês, *Option Trees*) na qual o principal benefício em relação às estratégias tradicionais de *ensemble* está em uma maior flexibilidade do tamanho das árvores. Visto que os métodos tradicionais podem gerar centenas de árvores similares com pequenas diferenças entre elas. Nesse sentido, a árvore de opções é capaz de representar essas diferenças em uma única estrutura de forma a economizar recursos computacionais como memória e processamento.

#### 5.4.2 Evolução de Dados

No contexto da aprendizagem *online* baseada na evolução dos dados, Bife et al. [36] propuseram uma versão modificada do algoritmo *Hoeffding Tree*, denominada de *Hoeffding Adaptive Tree* (HAT). A principal diferença entre o HAT e o *Hoeffding Tree* é que o HAT é capaz de lidar com as mudanças de conceitos nos dados ao longo do tempo. Isso significa que o modelo de classificação é atualizado somente quando uma mudança nos dados for detectada.

O algoritmo de detecção de mudanças no HAT deve sinalizar sempre que algo de importante acontece em cada nó da árvore de decisão. Nesse caso, o detector de mudanças é colocado para monitorar cada nó. Quando uma mudança é detectada em um determinado nó, o HAT gera uma nova árvore de decisão alternativa (sub-árvore) para representar o possível novo conceito. A sub-árvore é avaliada medida que ela for classificando novas amostras. Dessa forma, quando a sub-árvore alcança uma boa taxa de precisão ao longo do tempo, parte da árvore principal é substituída com a nova estrutura de dados da sub-árvore. Caso contrário, a sub-árvore é excluída da memória. Note que o HAT possui um processo de criação, gerenciamento, substituição e exclusão de árvores alternativas ao longo do treinamento do modelo de classificação.

Considere o seguinte cenário: Sejam  $C$  e  $D$  dois conceitos arbitrários que são representados por rótulos diferentes. Suponha que uma sequência  $S$  seja o dado de entrada para

o algoritmo HAT. Inicialmente, o conceito  $C$  aparece no tempo  $t_0$  e, abruptamente muda para o conceito  $D$  no tempo  $t_0 + 1$ , e permanece estável após isso. Após HAT rodar na sequência  $S$ , o erro  $e_1$  é obtido para o tempo  $t_0$  e o erro  $e_2$  é obtido para o tempo  $t_0 + 1$ , de modo que  $e_2 - e_1$  mede o quão pior é o erro após a mudança de conceito. Se  $|e_2 - e_1| > 2\epsilon_{cut}$ , então a mudança é detectada e o HAT começa a gerar a árvore alternativa. O valor de  $\epsilon_{cut}$  é calculado com base no algoritmo de detecção de mudanças ADWIN descrito a seguir.

O ADWIN (*ADaptive sliding WINdow*) [35] é um algoritmo de detecção de mudança baseado na análise de janelas de tempo organizadas de forma sequencial. A intuição por trás do ADWIN é simples. Dadas duas janelas de tempo adjacentes, se as médias entre elas forem diferentes o suficiente uma da outra, então o algoritmo conclui que houve mudança nos dados. Além disso, o ADWIN atua com um espaço de memória limitado, onde os dados antigos são descartados e os mais recentes são mantidos para as próximas comparações. A principal vantagem do ADWIN é caracterizada por ser um algoritmo livre de parâmetros sendo capaz de se adaptar a diferentes frequências de mudanças ao longo do tempo.

```

ADWINO: ADAPTIVE WINDOWING ALGORITHM
1  Initialize Window  $W$ 
2  for each  $t > 0$ 
3      do  $W \leftarrow W \cup \{x_t\}$  (i.e., add  $x_t$  to the head of  $W$ )
4          repeat Drop elements from the tail of  $W$ 
5              until  $|\hat{\mu}_{W_0} - \hat{\mu}_{W_1}| < \epsilon_{cut}$  holds
6                  for every split of  $W$  into  $W = W_0 \cdot W_1$ 
7                      output  $\hat{\mu}_W$ 

```

**Figura 5.3 Pseudocódigo do algoritmo ADWIN.**

Formalmente, o ADWIN atua sobre uma sequência  $X = \{x_1, x_2, \dots, x_t\}$ , na qual a sequência  $X$  pode ser subdividida em subsequências  $W = \{x_p, x_{p+1}, \dots, x_{p+w-1}\}$ , onde  $x_p$  representa um ponto da subsequência. Os dados de  $X$  devem ser normalizados em torno da média 0 e desvio padrão 1. Assim, seja  $\hat{\mu}_w$  a média conhecida esperada do valor  $x_p$ , o ADWIN percorre a janela  $W$  para ler o próximo valor  $x_{p+1}$  mais recente. Por outro lado, seja  $\mu_w$  a média desconhecida atual de  $\mu_t$  dentro de  $W$  com  $t \in W$ . Se a média  $\mu_w$  for suficientemente diferente de  $\hat{\mu}_w$ , então a mudança é sinalizada e a janela anterior é descartada. A fronteira de decisão da mudança é definida por *Hoeffding Bound* com base no valor de confiança  $\delta \in (0,1)$  [35].  $\delta$  define o nível de tolerância da mudança com base na escala da sequência original. O pseudocódigo da Figura 7.7 mostra o algoritmo ADWIN [212].

A linha 1 mostra a subsequência  $W$  como entrada de dados do algoritmo ADWIN. Em seguida, para cada amostra de  $X$  adiciona um  $x_t$  na cabeça de  $W$  (linha 2 e 3). Depois, para cada amostra de  $W$ , remove os elementos da calda até que a diferença nas médias  $\hat{\mu}_{w_0}$  e  $\hat{\mu}_{w_1}$  seja menor que o *Hoeffding Bound*  $\epsilon_{cut}$  (linhas 4 e 5). A linha 6 mostra  $W$  sendo dividido em  $W_0$  e  $W_1$ . Ambos possuem tamanhos  $n_0$  e  $n_1$ , respectivamente, sendo  $n = n_0 + n_1$ . Nesse ponto, que o  $\epsilon_{cut}$  é calculado com base na comparação entre as médias  $\hat{\mu}_{w_0}$  e  $\hat{\mu}_{w_1}$  de  $W_0$  e  $W_1$  e suas respectivas médias esperadas  $\mu_{w_0}$  e  $\mu_{w_1}$ . Assim, a equação 5.3 mostra a fórmula do  $\epsilon_{cut}$ .

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} * \ln(4/\delta')} \quad \text{Equação 5.3}$$

onde  $m$  significa a média harmônica de  $n_0$  e  $n_1$ ,  $m = \frac{1}{\frac{1}{n_0} + \frac{1}{n_1}}$ , e  $\delta' = \frac{\delta}{n}$ . Por fim, a linha 7 indica que a média  $\hat{\mu}_w$  de  $W$  é retornada como resultado do ADWIN. Note que  $\hat{\mu}_w$  é um valor aproximado de  $\mu_w$ . Mais detalhes sobre o ADWIN podem ser obtidos em [36].

Da mesma forma, o algoritmo HAT pode ser utilizado pelas estratégias de *Bagging* e *Boosting*. Neste caso, Bife et al. [36] também realizaram adaptações para que os *ensembles* reajam à mudança nos dados, ou seja, existem as variações *OzaBagAdwin* e *OzaBoostAdwin* que podem ser utilizados pelos algoritmos *online* adaptativos.

#### 5.4.3 Detecção de Novidades

No contexto da aprendizagem baseada na detecção de novidades, esta seção apresenta o algoritmo AnyNovel [26] utilizado como *baseline* neste trabalho. A vantagem do AnyNovel é que ele desenvolvido para resolver, especificamente, o problema de HAR baseado em smartphones instrumentados com sensores inerciais.

O AnyNovel foi proposto por Abdallah et al. [26] com o propósito de descobrir novos conceitos (ou atividades) ao longo do tempo e, como estratégia de consolidação do conceito, o algoritmo interage com usuário por meio da aprendizagem ativa. O AnyNovel é uma implementação e evolução de um *framework* proposto em um dos trabalhos anteriores de Abdallah et al. [25]. Basicamente, o *framework* é composto por três componentes. O primeiro componente atua na fase *offline*, onde o modelo de classificação é gerado com base em uma base de dados de treino inicial. O segundo e o terceiro componentes atuam na fase *online*, sendo que o segundo consiste no processo de classificação de novas instâncias e o terceiro consiste na adaptação do modelo de classificação. O componente de adaptação é utilizado quando os conceitos evoluem (*incremental learning*) ou quando surgem novos conceitos ao longo do tempo.

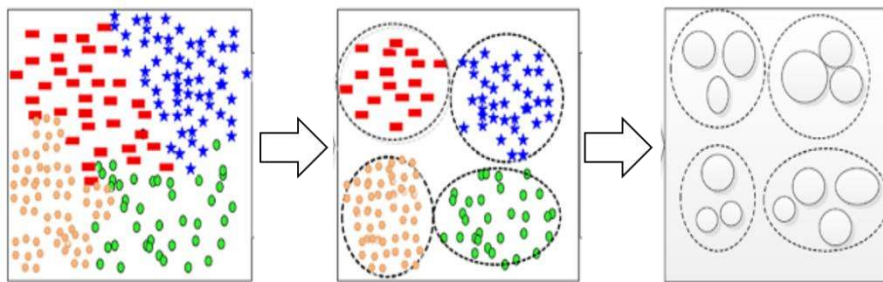


Figura 5.4 Intuição de como funciona o algoritmo AnyNovel.

A Figura 5.4 mostra que o algoritmo AnyNovel é baseado no agrupamento de dados. Cada rótulo (ou atividade) é representado por um *cluster* e, cada *cluster* possui um conjunto de *micro clusters* que representam conceitos secundários dentro de um *cluster*. Por exemplo, no contexto de HAR um *micro cluster* pode representar condições particulares de uma atividade, como andar (*cluster*) e andar lento (*micro cluster*). O *micro cluster* permite que o AnyNovel seja capaz de descobrir novos conceitos tanto fora quanto dentro de um *cluster*, contribuindo para que o algoritmo consiga aprender detalhes sobre as atividades humanas.

Os *clusters* e *micro-clusters* são definidos com base em medidas estatísticas, como centroides, pesos, raios e distâncias entre *clusters*, conforme está descrito na Tabela 5.1. Para facilitar o entendimento, a Figura 5.5 mostra um esquema geral de como os *clusters* são distribuídos no modelo de classificação. Como pode ser observado, cada *cluster* é representado

por uma circunferência que possui um centroide  $C$  e é delimitada por um raio  $R$ . Os novos dados são acumulados na memória através de um *buffer*. O *buffer* é composto por dados de vários *chunks* que são equivalentes as janelas de tempo. À medida que novos *chunks* são coletados, o AnyNovel calcula as distâncias entre os centroides dos dados do *buffer* e *chunk* e os centroides dos *clusters*. Além disso, cada cluster possui um faixa de tolerância (DRAB) utilizado para monitorar as mudanças de conceitos dos rótulos ao longo do tempo.

Tabela 5.1 Características do modelo do AnyNovel.

Symbol	Level	Description
Centroid	Micro Cluster	Média dos $n$ dimensionais pontos dentro do micro cluster
Weight		Total de dados dentro do micro cluster
Radius		Distância máxima entre o centroide e todos as instâncias de dados dentro do micro cluster
Nº Sub-clusters	Cluster	Número de micro clusters dentro do cluster
Centroid		Média de todos os centroides dos micro clusters dentro do cluster
Weight		Total de dados dentro do cluster
Radius	Global	Distância máxima entre o centroide e todos as instâncias de dados dentro do cluster
Max distance		Máxima distância entre qualquer par de clusters
Nº Clusters		Total de clusters ou atividades do domínio
Centroid		Centroide global de todas as instâncias da base de dados de treino
Capacity		Total de instâncias da base de dados de treino

O componente de adaptação do AnyNovel é composto por duas funcionalidades importantes. A primeira consiste na avaliação da coesão e da consistência dos dados contidos no *buffer* por meio do *Cohesion Validation Component* (CVC). A segunda consiste em decidir se os dados do *buffer* são conhecidos, desconhecidos ou é uma novidade por meio do componente OBSERVER.

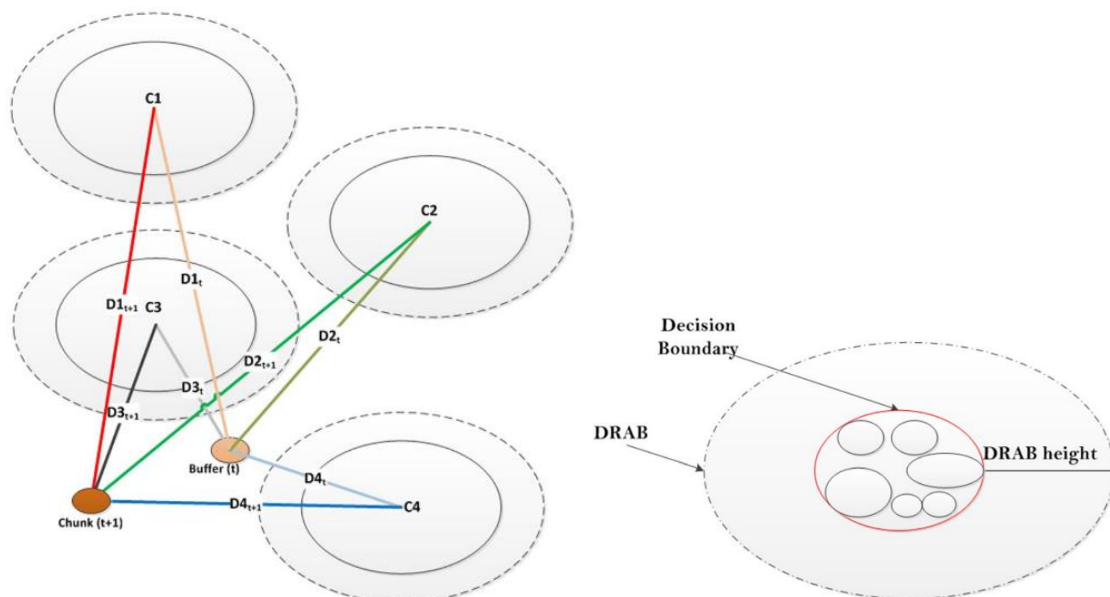


Figura 5.5 Exemplo da formação de clusters pelo algoritmo AnyNovel (lado esquerdo) e a estrutura interna de um cluster e micro cluster (lado direito).

O objetivo do componente CVC é monitorar a ocorrência de mudanças nos dados e tratar *outliers*, avaliando a dependência dos dados entre o *buffer* e o modelo de classificação. O CVC é baseado em 4 operações: *seed*, *replace*, *reset*, and *keep*. A operação *seed* significa que o *chunk* mais recente é coerente com a última classificação dos dados. A operação *replace* significa que um *outlier* foi detectado. A operação *reset* significa que os dados do *chunk* e *buffer* são complementemente diferentes. A operação *keep* significa que os dados do *chunk* e *buffer* são coerentes.

O componente OBSERVER monitora a evolução dos dados entre o *buffer* e os *chunks*. Basicamente, o OBSERVER verifica se os dados do *buffer* estão dentro dos limites de um *cluster*, incluindo a faixa de tolerância (DRAB) mostrada na Figura 5.5. Nesse caso, existem três possíveis decisões: se os dados do *buffer* estiverem dentro do *cluster*, então os dados são conhecidos. Se os dados do *buffer* estiverem dentro da faixa de tolerância, então isso representa uma evolução ou mudança de conceito. Por fim, se os dados do *buffer* estiverem completamente fora do *cluster*, então os dados são desconhecidos.

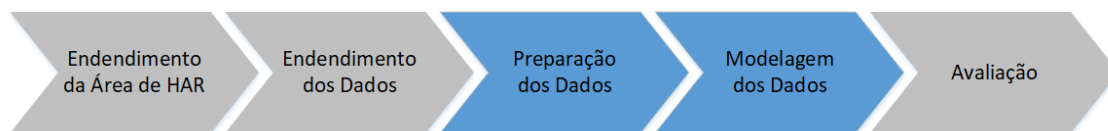
Além disso, o AnyNovel possui um mecanismo de esquecimento de conceitos baseado no tempo. A regra para isso é simples, se um determinado *cluster* do modelo não estiver sido utilizado para classificar uma atividade por um determinado tempo pré-definido, então esse *cluster* é removido do modelo. A remoção do *cluster* é simplesmente a remoção das características fundamentais descritas na Tabela 5.1. Mais detalhes sobre o AnyNovel podem ser obtidos em [26].

## 5.5. Considerações Finais

Este capítulo mostrou os conceitos dos algoritmos *online* destacando três grupos: 1) os algoritmos de aprendizagem cega; 2) os algoritmos baseados na evolução de dados; e 3) os algoritmos baseados em detecção de novidades. Além disso, este capítulo também descreveu os *baselines* utilizados na comparação com o algoritmo simbólico adaptativo NOHAR, proposto por esta tese de doutorado. Mais detalhes sobre os algoritmos *online* adaptativos podem ser obtidos em [35]. Além disso, este capítulo contém os conceitos necessários que são base para o método proposto por esta tese de doutorado.

# CAPÍTULO 6

## MÉTODO ONLINE SIMBÓLICO PARA HAR



Este capítulo apresenta a proposta principal desta tese de doutorado composta pelo o *framework* DISTAR (*DIScrete Stream learning for Activity Recognition*) e pelo algoritmo *online* simbólico NOHAR (*NOvelty discrete data stream for Human Activity Recognition*). Nossa proposta é baseada em uma nova estratégia relacionada à análise de fluxo contínuo de dados discretos. Inicialmente, são apresentados os motivos pelos quais este caminho foi escolhido. Em seguida, são apresentados os detalhes referentes ao *framework* DISTAR e ao algoritmo NOHAR.

### 6.1. Fundamentos da Proposta

O capítulo anterior mostrou a fundamentação teórica e as vantagens da aprendizagem *online* na área de HAR. Como foi visto, a natureza do problema de reconhecimento de atividades humanas torna o uso das técnicas de aprendizagem *online* uma ótima opção para descobrir e reconhecer as atividades em tempo real. Além disso, a natureza da coleta contínua dos dados dos sensores inerciais também contribui para a escolha dessa estratégia.

O Capítulo 5 apresentou os algoritmos *online* AnyNovel [26], *Hoeffding Tree* [213] e o *Hoeffding Adaptive Tree* [36] que podem ser empregados no desenvolvimento de sistemas HAR. Embora esses algoritmos possam ser utilizados para reconhecer padrões de comportamento humano, ainda existe uma desvantagem relacionada ao custo computacional do processamento de séries temporais compostas por valores contínuos, em contraste com o processamento de séries temporais discretizadas. Um olhar atento à literatura mostra que ainda não existe um algoritmo de aprendizagem *online* adaptado para manipular dados simbólicos. Nesse sentido, uma hipótese é que o surgimento de algoritmos *online* baseados em algoritmos de representação simbólica pode minimizar o processamento de dados, tornando as respostas mais rápidas e eficientes. Gama et al. [35] sugerem o uso da estratégia de representação simbólica no contexto de *data streaming*, inclusive com a aplicação dos algoritmos SAX e PiD. O Capítulo 4 mostrou várias soluções de discretização de séries temporais que podem ser utilizadas na área de HAR. Os resultados dos experimentos descritos na Seção 7.3 mostram que as soluções baseadas nas características do domínio discreto funcionam muito bem no contexto da aprendizagem *offline*, tendo resultados superiores às estratégias tradicionais que, inclusive, manipulam valores contínuos. Dessa forma, é esperado que essa estratégia também possa funcionar na perspectiva dos algoritmos de aprendizagem *online* mantendo, assim, a eficiência dos modelos de classificação.

Neste contexto, esta proposta de tese de doutorado introduz um novo algoritmo de aprendizagem *online* pioneiro na manipulação de dados simbólicos. O objetivo é, além de preencher a lacuna da falta desse tipo de algoritmo na área de *data streaming*, desenvolver um

algoritmo de baixo custo capaz de lidar com as limitações dos dispositivos móveis como processamento, memória e, conseqüentemente, consumo de energia. O algoritmo proposto combina as vantagens da área de aprendizagem *online* com as vantagens das técnicas de representação simbólica. Essa combinação tem motivado e direcionado esta tese de doutorado para o desenvolvimento de uma solução genérica que pode ser aplicada não somente na área de HAR, mas também em outras áreas que possuem características similares a esta.

Portanto, o método proposto por esta tese de doutorado é composto por um *framework* denominado DISTAR (*DI*screte *ST*ream learning for *ACT*ivity *RE*cognition) e um algoritmo *online* adaptativo denominado NOHAR (*NO*velty discrete data stream for *HU*man *ACT*ivity *RE*cognition). O *framework* DISTAR é um *framework* adaptado para suportar algoritmos de representação simbólica na perspectiva da abordagem de *data streaming*. O algoritmo NOHAR é um algoritmo *online* da categoria de detecção de novidade.

## 6.2. Framework DISTAR

O *framework* DISTAR atua sobre uma quantidade limitada de memória, aqui denominada de *buffer*. O *buffer* consiste em uma pequena unidade de memória volátil com o propósito de armazenar uma quantidade limitada de dados. Nesse caso, o *buffer* pode ser dividido em unidades de memória menores denominadas de *chunks*. Cada *chunk* é representado por uma janela de tempo. No contexto dos dados simbólicos, cada *chunk* é transformado em palavras por meio do processo de discretização. Em seguida, as palavras são organizadas em termos de suas frequências. Visualmente, a distribuição de frequência é representada por um histograma.

No *framework* DISTAR, os histogramas são vistos como *Bag-of-Pattern* (BOP). Um BOP pode conter informações suficientes de atividades e ações dos usuários. Nesse caso, o papel do algoritmo de aprendizagem é analisar, detectar e reconhecer padrões nos BOPs gerados a partir do fluxo contínuo de dados. O DISTAR classifica os BOPs em três categorias: i) *Unknown* BOP (uBOP) – quando os histogramas são desconhecidos; ii) *Novel* BOP (nBOP) – quando os histogramas apresentam alguma novidade ou um novo conceito nos dados; e iii) *Reference* BOP (rBOP) quando o histograma é conhecido pelo modelo de classificação. A Figura 6.1 mostra uma visão geral de como ocorre a discretização e a geração de histogramas previstas pelo *framework* DISTAR.

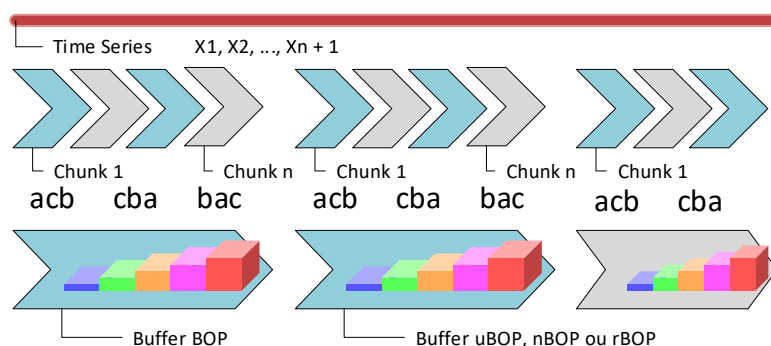


Figura 6.1 Visão geral de como ocorre a discretização dos dados e a geração de BOPs.

A Figura 6.2 mostra uma visão geral das etapas de processamento de dados do *framework* DISTAR. A primeira etapa consiste na discretização dos dados brutos. A segunda etapa consiste na geração dos histogramas BOPs. A terceira etapa consiste na classificação e



discriminação das atividades humanas. Por fim, a quarta etapa consiste na aprendizagem das atividades por meio da análise dos BOPs. No DISTAR, cada histograma representa apenas uma atividade e, cada atividade, pode ser representada por um ou mais histogramas.

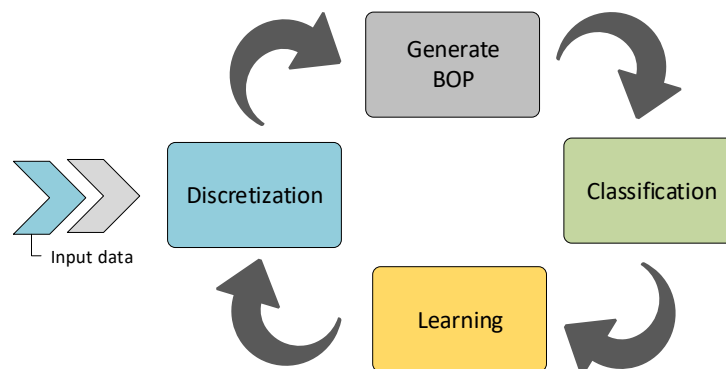


Figura 6.2 Etapas do framework DISTAR.

A Figura 6.3 mostra uma visão mais detalhada das etapas do *framework* DISTAR. A etapa de discretização tem como objetivo compactar os dados brutos com o propósito de reduzir a dimensionalidade dos dados e otimizar o uso da memória física dos dispositivos. Para isso, qualquer discretizador pode ser utilizado como, por exemplo, os algoritmos de representação simbólica SAX [38], iSAX [48] ou SFA [39]. A segunda etapa consiste em organizar os dados discretizados na forma de BOPs. Cada BOP pode representar um padrão (ou não) que deve ser tratado na etapa de classificação e aprendizagem. A etapa de classificação consiste na discriminação dos BOPs gerados a partir do fluxo contínuo de dados em uBOP, nBOP ou rBOP.

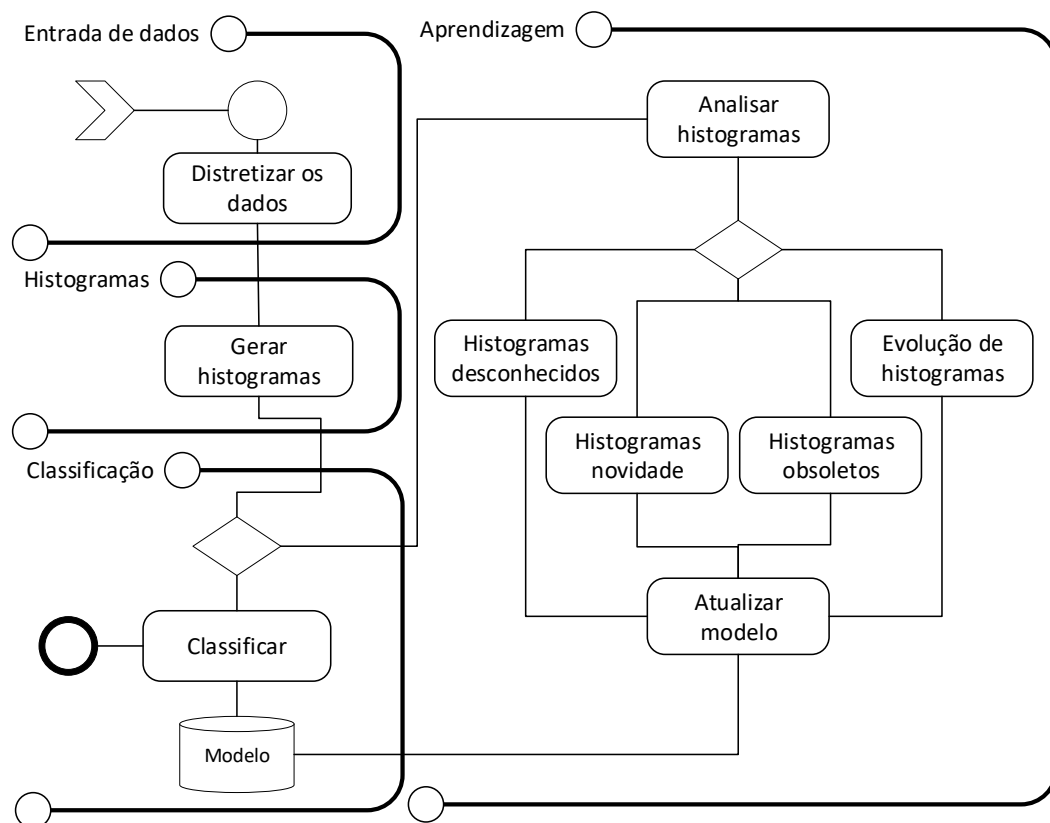


Figura 6.3 Visão mais detalhada das etapas do framework DISTAR.

A etapa de aprendizagem consiste em uma análise mais aprofundada dos histogramas uBOP, nBOP ou rBOP. Para todos eles, o algoritmo verifica se há alguma alteração na distribuição de frequência dos histogramas causada, por exemplo, por alguma alteração no comportamento do usuário. Essa alteração é conhecida na área de aprendizagem *online* como evolução ou mudança de conceitos (*Concept Drift*). Quando o algoritmo detecta uma mudança na distribuição dos dados, o modelo de classificação é atualizado. Na prática, o algoritmo compara se as distribuições de frequência entre dois histogramas são similares, isso pode levar a três caminhos: i) se a taxa de similaridade estiver dentro de um limiar de confiança definido como parâmetro, então não há mudança nos dados; ii) se a taxa de similaridade estiver fora do limiar de confiança, então os histogramas são diferentes; iii) se a taxa de similaridade estiver entre o limiar de confiança e o limiar de tolerância, então pode estar ocorrendo uma evolução nos dados. O limiar de tolerância permite identificar mudanças de conceitos nos histogramas uBOP, nBOP e rBOP. Portanto, é necessário que o histograma de referência receba um ajuste na frequência das palavras com o propósito de se adaptar as mudanças ao longo do tempo. A Seção 6.3.3 mostra mais detalhes de como é tratada a evolução de conceito em histogramas.

A etapa de aprendizagem também é responsável por controlar o ciclo de vida dos histogramas. Por exemplo, um uBOP pode se tornar um nBOP se a aparição desse histograma for recorrente ao longo do tempo e a frequência de repetições ultrapassar um limiar definido como parâmetro. Da mesma forma ocorre com o nBOP. A diferença é que o usuário é requisitado para rotular o histograma por meio da aprendizagem ativa. Quando a frequência de repetições de um nBOP atinge um limiar, esse histograma é transformado em um rBOP e adicionado ao modelo de classificação. Nesse caso, a aprendizagem ativa não é acionada.

Além disso, o DISTAR é capaz de detectar atividades obsoletas, ou seja, atividades que não são mais executadas pelo usuário. Na prática, os histogramas que representam essas atividades podem ser excluídos do modelo. A Seção 3.3 também mostra detalhes técnicas de como isso ocorre.

### 6.3. Algoritmo NOHAR

Esta seção descreve detalhes do algoritmo NOHAR (*NOvelty discrete data stream for Human Activity Recognition*) que é uma instância do *framework* DISTAR. Como pode ser observado, o *framework* DISTAR estabelece um conjunto de etapas genéricas que podem ser adaptadas para diversos cenários com o uso de diferentes técnicas especializadas em cada etapa. O algoritmo NOHAR, por sua vez, é um exemplo dessa adaptação voltada para o cenário da análise de séries temporais. Este cenário foi definido com base na natureza dos dados dos sensores inerciais formados pelas coordenadas  $x, y$  e  $z$ . A Figura 6.4 mostra detalhes das tarefas executadas nas três fases do algoritmo NOHAR por meio de um diagrama BPMN 2.0<sup>2</sup> (*Business Process Model and Notation*). As próximas seções apresentam mais detalhes sobre cada uma dessas etapas.

A fase de discretização dos dados ocorre em quatro etapas. Na primeira, os dados são segmentados para a formação do *chunks* (ver Seção 6.3.1.1). Na segunda, os dados correspondentes a cada *chunk* são pré-processados com os filtros *Exponential Smoothing* e *Lowpass* [216] com o propósito de reduzir ruídos causados, por exemplo, pela transição entre atividades (ver Seção 6.3.1.1). Na terceira, os *chunks* são submetidos ao processo de discretização utilizando o algoritmo SAX (ver Seção 6.3.1.2). Por fim, as informações estatísticas dos *chunks* (e.g. média e variância) são extraídas e vinculadas às palavras

<sup>2</sup> <http://www.bpmn.org/>

correspondentes a cada *chunk*. Essa estratégia tem como objetivo fornecer insumo ao modelo de classificação na tentativa de melhorar a discriminação das atividades humanas.

A fase de geração e análise dos histogramas é composta por três etapas. Na primeira, o alinhamento das palavras é verificado com base na consulta a um dicionário que contém todas as palavras que representam um mesmo segmento em diferentes formas de deslocamento (ver Seção 6.3.2). O objetivo é diminuir a dimensionalidade dos histogramas evitando o uso de palavras deslocadas. Em seguida são realizadas as operações para criar um novo histograma, adicionar novas palavras a um histograma e atualizar a frequência das palavras no histograma. Por fim, as palavras contidas nos histogramas são ordenadas com o propósito de facilitar a comparação de histogramas na fase de aprendizagem.

A fase de aprendizagem do algoritmo NOHAR é composta por 7 etapas. A primeira tarefa consiste em buscar os histogramas similares ao novo histograma gerado após o processamento do fluxo contínuo de dados. A busca por histogramas similares ocorre nos *buffers* uBOP, nBOP e rBOP. A ordem das buscas é a seguinte: 1) o NOHAR verificar se existe histograma similar no *buffer* rBOP, se sim, o histograma é classificado e a busca é finalizada; 2) o NOHAR verifica se existe histograma similar no *buffer* nBOP, se sim, o histograma é classificado, a busca é finalizada e o NOHAR decide se transforma (ou não) o nBOP em rBOP; 3) o NOHAR verifica se existe histograma similar no *buffer* uBOP, se sim, o NOHAR decide se transforma (ou não) o uBOP em nBOP. Caso exista mais de um histograma similar, o NOHAR seleciona o histograma uBOP, nBOP ou rBOP mais próximo ao histograma não rotulado, ou seja, seleciona o histograma de menor distância (ver Seção 6.3.3).

Além disso, o NOHAR monitora os histogramas que não são mais utilizados (ou discriminados) pelo modelo de classificação. Desse modo, se determinado histograma ficar obsoleto, o NOHAR elimina-o dos *buffers* nBOP ou rBOP. O NOHAR também monitora a evolução e mudanças de conceitos nos histogramas observando os limiares de confiança e tolerância descritos na Seção 6.3.3.

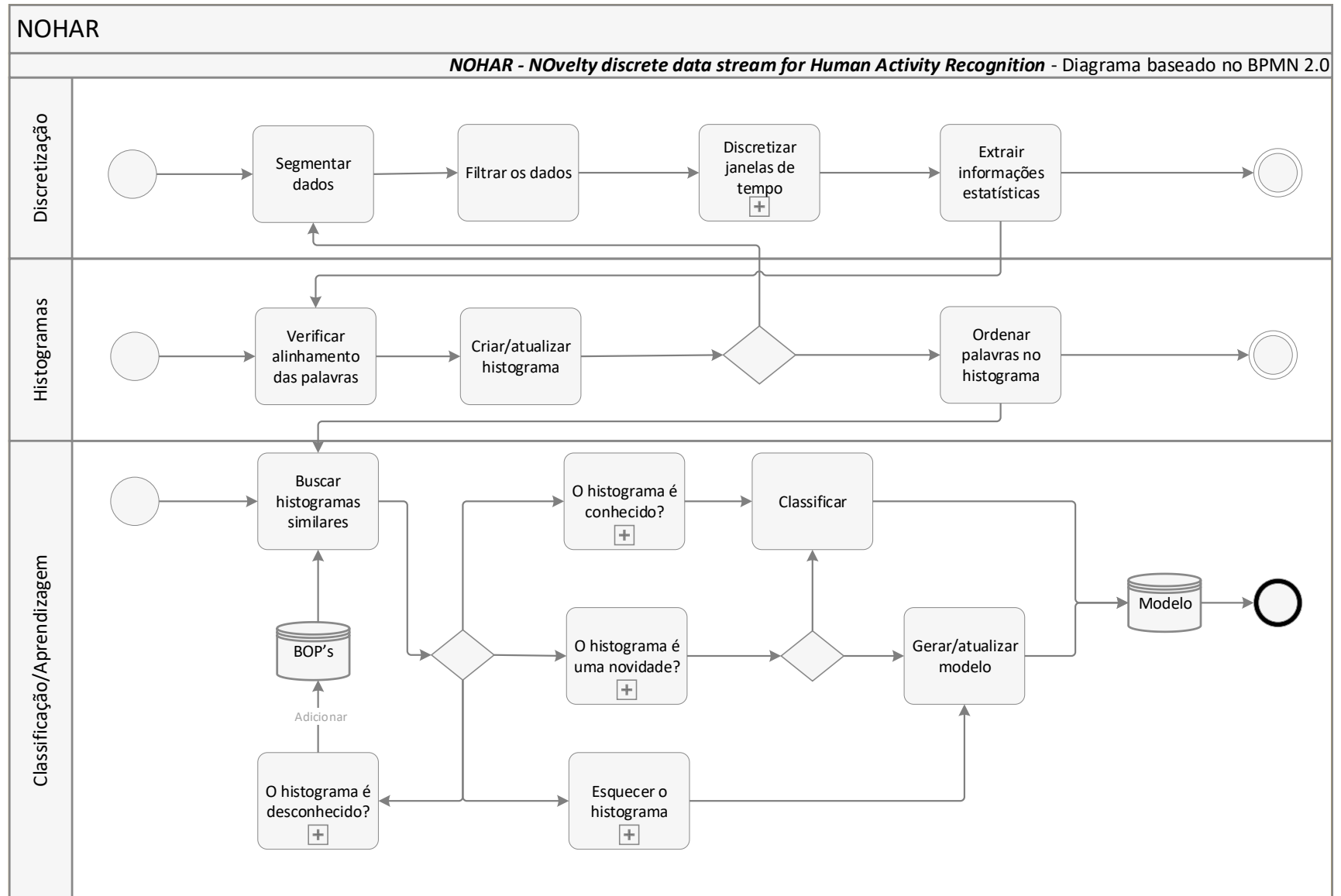


Figura 6.4 Diagrama BPMN que detalha o passo a passo da algoritmo NOHAR.

### 6.3.1 Discretização

O algoritmo NOHAR é projetado para lidar com dados brutos de múltiplas séries temporais. O processamento de dados em cada série temporal é realizado de forma sincronizada (ou paralela) e, como resultado, os dados simbólicos (palavras) gerados são reunidos em um único histograma. O processo de discretização é executado nos dados contidos em um *chunk*. As próximas seções descrevem como funciona a segmentação dos dados brutos e o processo de discretização.

#### 6.3.1.1 Entrada de Dados e Segmentação

Neste trabalho, as séries temporais são extraídas dos sensores inerciais conforme mostra o exemplo exibido na Figura 6.5. Como pode ser observado, os dados dos sensores inerciais são formados pelas coordenadas  $x$ ,  $y$  e  $z$ . Cada coordenada forma uma série temporal. Nesse caso, os dados da série temporal representam a atividade ‘Walk’.

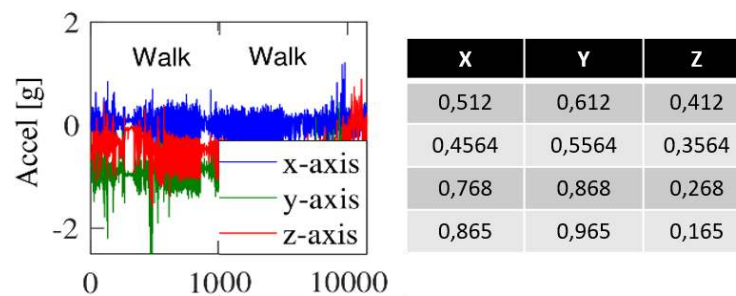


Figura 6.5 Entrada de dados do NOHAR.

O primeiro passo do algoritmo NOHAR é segmentar os dados em janelas de tempo de tamanhos fixos (*chunks*). Em seguida, os dados de cada *chunk* são transformados pelos filtros *Exponential Smoothing* e *Lowpass* [216]. O objetivo de aplicar um filtro é tentar melhorar a qualidade dos dados por meio da redução de ruídos nos sinais. No contexto dos algoritmos de representação simbólica, quanto menor a presença de ruídos, maior a consistência dos dados discretos.

O NOHAR utiliza filtro alisamento exponencial (*Exponential Smoothing*). A vantagem desse filtro é que ele dá mais pesos às informações mais recentes da série temporal. Com isso, é possível realçar mudanças bruscas no sinal de forma mais rápida evitando o atraso causado, por exemplo, pelo filtro de média móvel. Formalmente, o filtro é aplicado sobre uma sequência  $x_t$  começando no tempo  $t = 0$ . A saída é representada por um estimador  $s_t$  que indica o próximo valor de  $x$  no próximo tempo  $t$ . A fórmula do filtro é representada pela equação 6.1:

$$s_t = \alpha x_t + (1 - \alpha)x_{t-1}, \quad t > 0 \quad \text{Equação 6.1}$$

onde  $\alpha$ ,  $0 < \alpha < 1$ , é um fator de suavização que determina o nível de aproximação do valor estimado ao valor real. Quanto maior for o  $\alpha$ , mais próximo do sinal original o resultado de  $s_t$  será. Além disso, o NOHAR utiliza uma variação do  $\alpha$  baseada no filtro *Lowpass* conforme mostra a equação 6.2:

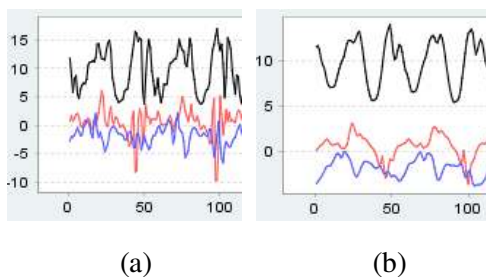
$$\alpha = e^{-2\pi(\frac{1}{f})} \quad \text{Equação 6.2}$$

onde  $e$ ,  $e \approx 2.718$ , é uma constante conhecida como número de Euler que representa a base de logarítmica natural e  $f$  representa a frequência do sensor inercial em Hertz. A intuição por trás desse  $\alpha$  é, basicamente, melhorar o resultado de  $s_t$  com uma informação adicional de

sazonalidade baseada na frequência do sensor. Por fim, o sinal é novamente processado utilizando a equação 6.3:

$$s_t = ax_t + ax_{t-1} \quad \text{Equação 6.3}$$

A Figura 6.6 mostra um exemplo do resultado do sinal de um sensor inercial após a aplicação do filtro do NOHAR. Como pode ser observado, o filtro consegue suavizar o sinal, capturar a sazonalidade e as tendências de forma eficiente. Desse modo, a propriedade do sinal é mantida sem perder muita informação nessa etapa inicial.



**Figura 6.6 (a) representa os dados originais e (b) representa os dados após a aplicação do filtro Exponential Smoothing. OBS: Linha preta é o eixo x, linha azul é o eixo y e linha vermelha é o eixo z.**

### 6.3.1.2 Processo de Discretização dos Dados

A próxima tarefa é a discretização do sinal. Para isso, o NOHAR utiliza o algoritmo SAX descrito na Seção 3.5.2.1.

Teoricamente, o SAX é o algoritmo mais adequado para atuar no contexto da aprendizagem *online*, quando comparado ao algoritmo SFA. O problema do SFA é a sua dependência ao treinamento prévio do MCB (*Multiple Coefficient Bin*). No contexto de *data streaming*, os dados mudam constantemente ao longo do tempo. Com isso, o SFA pode assumir diferentes MCB para dados similares conforme novos conceitos forem surgindo ao longo do tempo. Portanto, a desvantagem é que as palavras geradas para um determinado rótulo no tempo  $t$  pode ser diferente das palavras geradas para o mesmo rótulo no tempo  $t + n$ . Isso pode dificultar a aprendizagem dos algoritmos de classificação dependentes do SFA.

O tratamento de múltiplas séries temporais pelo algoritmo NOHAR é baseado no enriquecimento das palavras por meio de informações de contexto. Tais informações são baseadas em propriedades estatísticas das sequências originais que geram as palavras. A motivação para o enriquecimento das palavras por meio de informações de contexto é fundamentada no impacto causado pela normalização de dados utilizada pelo algoritmo SAX. Basicamente, essa normalização transforma a média de todas as séries temporais em torno de 0. O problema é que quando se trata da análise dos dados de múltiplas séries temporais, a falta dessa informação pode gerar palavras iguais para séries temporais diferentes. Dessa forma, para os problemas onde a média da série temporal é importante, os modelos de classificação podem ter dificultados na distinção das classes (ou atividades).

No contexto de HAR existem diferentes atividades onde o que diferencia elas umas das outras é a informação da média dos segmentos. Por exemplo, a Figura 6.7 mostra um exemplo onde o sinal da esquerda representa a atividade *sitting*, com a média dos eixos sendo 9, 3 e 1.8. O sinal da direita representa a atividades *standing*, com a média dos eixos 8, 6 e 2. Além disso, todos os sinais são parecidos uma vez que eles possuem variância e desvio padrão similares.

Nesse caso, o algoritmo SAX gera a mesma palavra para todos eles devido o processo de normalização transformar as médias de todas essas sequencias em torno de 0.

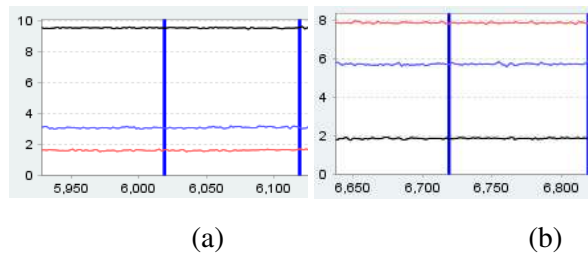


Figura 6.7 (a) representa os dados das atividades, sitting e (b) os dados da atividade standing. Esses dados foram extraídos da base de dados UCI-HAR. OBS: Linha preta é o eixo x, linha azul é o eixo y e linha vermelha é o eixo z.

Para resolver esse problema, esse trabalho propõe a associação das informações estatísticas, média e variância, da sequência original às palavras geradas após o processo de discretização. Portanto, para toda palavra  $p$  existe um contexto  $c$  associado. Assim, o contexto de uma palavra  $c(p)$  é definido pela equação 6.4:

$$c(p) = \frac{1}{n} \sum_{k=0}^n x_k + \frac{1}{n} \sum_{k=0}^n (x_k - \frac{1}{n} \sum_{k=0}^n x_k)^2 \quad \text{Equação 6.4}$$

onde  $x_k$  é o valor da sequência na posição  $k$ . Portanto, cada  $c(p)$  é associado a sua respectiva palavra. Além disso, cada palavra também é associada com sua respectiva coordenada.

A Figura 6.8 mostra um exemplo do resultado do processamento da etapa de discretização. Nesse caso, cada palavra é identificada com sua respectiva coordenada e as frequências são acumuladas com base no somatório da média e variância definida na Equação 6.4. A próxima seção mostra como o  $c(p)$  é utilizado na fase de criação dos histogramas.

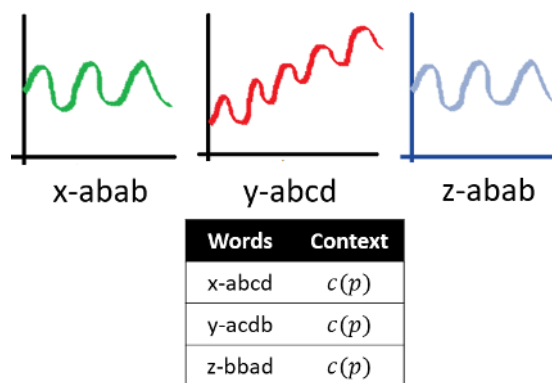


Figura 6.8 Exemplo do resultado da etapa de discretização.

### 6.3.2 Geração dos Histogramas de Palavras

O processo de geração dos histogramas de palavras, na técnica NOHAR, é simples e ocorre em três etapas.

A primeira etapa consiste na detecção de palavras desalinhadas. Palavras desalinhadas significam que uma única estrutura da série temporal pode ser representada por mais de uma palavra. Isso ocorre porque a posição dos intervalos das janelas de tempo pode ser deslocada ao longo da série temporal de maneira que as estruturas similares podem ser afetadas com a

geração de palavras diferentes. Por exemplo, a palavra “*abcd*” pode ser considerada idêntica à “*dabc*”, com as letras “*abc*” deslocadas à direita e a letra “*d*” deslocada ao início da cadeia de caracteres. Os deslocamentos de palavras são comuns ao longo da série temporal por conta de possíveis ruídos ou transição entre atividades. Portanto, o tratamento de palavras desalinhadas permite a redução ainda mais da dimensionalidade de dados nos histogramas, visto que palavras deslocadas podem ser consideradas palavras iguais. Dessa forma, os histogramas ficam mais “enxutos” em termos de variedade de palavras, isso aumenta a velocidade do processamento de dados nas etapas posteriores. A Figura 6.9 mostra um exemplo desse comportamento em uma série temporal homogênea.

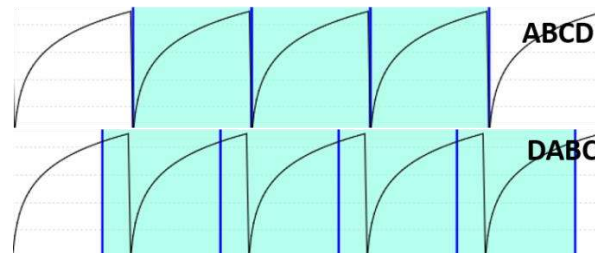


Figura 6.9 Exemplo de deslocamento de palavras.

A estratégia utilizada para tratar o alinhamento de palavras consiste na geração de um dicionário com todas as combinações das palavras existentes no histograma. O processo de verificação consiste em analisar todas as palavras do histograma e verificar a existência de palavras desalinhadas com base no dicionário. O tamanho do dicionário é igual à  $n - 1$ , onde  $n$  é o tamanho da palavra. Por exemplo, a palavra de tamanho “*abcd*” tem três variações “*dabc*”, “*cdab*” e “*bcda*”. As variações ocorrem em função do deslocamento das letras para esquerda ou para direita.

A segunda etapa consiste em definir a quantidade de *chunks* responsáveis por gerar um histograma. Na prática, o que se define é o intervalo de tempo da série temporal original no qual será representado pelo histograma. Além disso, é necessário definir a quantidade de pulos (*offset*) que serão dados durante o processo de segmentação da série temporal. Por convenção, o *offset* é definido pela razão entre o tamanho do *chunk* e o tamanho da palavra. Essa razão é equivalente a pular os dados de uma subsequência do tamanho de uma letra. Isso evita que o histograma seja formado por um alto índice de palavras desalinhadas.

A atualização do histograma consiste em: 1) primeiro detectar as palavras desalinhadas e escolher uma delas; 2) atualizar a frequência da palavra existente ou adicionar a nova palavra ao histograma. O processo de atualização de frequência é baseado na equação 6.5:

$$f(p) = f(p) + c(p) \quad \text{Equação 6.5}$$

onde  $f(p)$  é a frequência atual da palavra e  $c(p)$  é o contexto da palavra definido na etapa de discretização. A inclusão do contexto da palavra na distribuição de frequência do histograma contribui para melhorar a discriminação de classes (ou atividades) do modelo de classificação gerado pelo NOHAR.

Por fim, a terceira tarefa consiste em ordenar as palavras do histograma com o propósito de agilizar o cálculo da distância e a fusão de histogramas nas etapas posteriores. Palavras ordenadas aumentam a velocidade de busca das palavras durante a comparação entre dois histogramas. A Figura 6.10 mostra um exemplo do resultado final da etapa de geração de histogramas do algoritmo NOHAR.



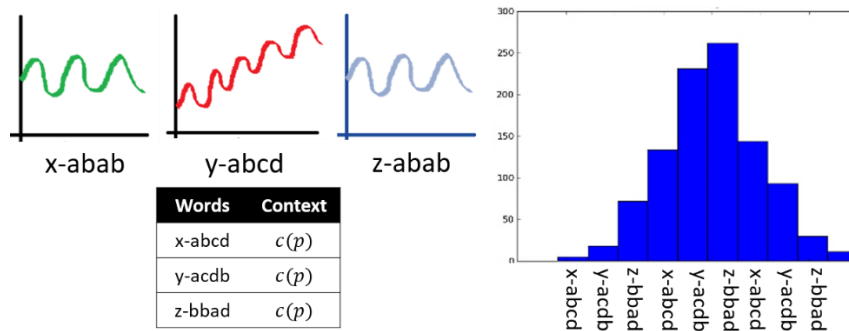


Figura 6.10 Exemplo do resultado final da etapa de geração de histogramas.

### 6.3.3 Aprendizagem

A etapa de aprendizagem do algoritmo NOHAR corresponde ao componente de adaptação do *framework* DISTAR. Esta etapa consiste na análise dos histogramas não rotulados gerados ao longo do tempo a partir do fluxo contínuo de dados. Intuitivamente, cada atividade humana é representada por um ou mais histogramas (nBOP e rBOP). O conjunto de todos rBOP compõe o modelo de classificação.

A primeira tarefa do processo de aprendizagem consiste em recuperar todos os histogramas similares ao novo histograma não rotulado. A ideia é identificar se o histograma atual corresponde a um padrão já existente (rBOP), se corresponde a um novo padrão (nBOP) ou se corresponde a um padrão desconhecido (uBOP).

A literatura apresenta várias técnicas para medir a semelhança entre histogramas baseadas em cálculos de distâncias, como a distância euclidiana cumulativa (DEC), a distância Land Mover (DLM) e a distância modular [217]. O NOHAR utiliza um cálculo de distância baseada em distribuição de frequências proposta por Schäfer [42]. No entanto, essa técnica sofreu adaptações com o propósito de resolver o problema de desigualdade triangular e assimetria da distância entre as distribuições de frequências de dois histogramas.

Basicamente, o problema da desigualdade triangular consiste em: dado dois histogramas,  $BOP_1$  e  $BOP_2$ , a distância entre  $BOP_1$  e  $BOP_2$  é diferente da distância entre  $BOP_2$  e  $BOP_1$ , exceto quando ambos são iguais. Esse problema afeta a verificação do vizinho mais próximo, pois quando  $BOP_1$  é o mais próximo de  $BOP_2$ , não significa que  $BOP_2$  é o mais próximo de  $BOP_1$ . A Figura 6.11 mostra um exemplo de dois histogramas claramente diferentes um do outro. O resultado do cálculo da distância entre eles é  $dist(BOSS_1, BOSS_2) = 9$  e  $dist(BOSS_2, BOSS_1) = 14$ .

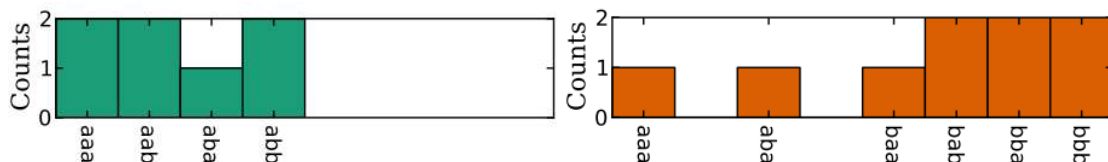


Figura 6.11 Exemplo de dois histogramas BOSS.

A fórmula da distância utilizada por Schäfer [42] é representada pela equação 6.6:

$$dist(BOSS_1, BOSS_2) = \sum_{k=1}^n (BOSS_1(f(p_k)) - BOSS_2(f(p_k)))^2_{BOSS_1(f(p_k)) > 0} \quad \text{Equação 6.6}$$

onde,  $BOSS_1$  e  $2$  são os histogramas a serem comparados,  $f(p_k)$  é a frequência da palavra  $p_k$  e  $n$  é a quantidade de palavras no histograma.

O primeiro passo para combater a assimetria é estabelecer uma regra para tratar dois histogramas de forma unilateral. Isso pode ser feito, por exemplo, sempre comparando o maior histograma em relação ao menor histograma. Nesse caso, o tamanho do histograma é medido com base na quantidade de palavras  $p = (p_1, p_2, \dots, p_n)$ , onde  $n$  representa o número de palavras. Portanto, a condição necessária para medir a distância entre o  $BOP_1$  e o  $BOP_2$  é  $size(BOP_x) > size(BOP_y)$ , onde  $BOP_x$  pode ser representado pelo  $BOP_1$  ou  $BOP_2$  dependendo de quem for o maior.

O segundo passo para resolver a assimetria é levar em consideração as palavras do  $BOP_x$  que não estão contidas no  $BOP_y$ , ou seja,  $BOP_x(p) \notin BOP_y(p)$ . A estratégia utilizada pelo BOSS ignora esse conjunto de palavras atribuindo o valor '0' durante o cálculo da distância. O problema é que essas palavras podem influenciar no quão diferente dois histogramas podem ser. No NOHAR, as informações desse conjunto de palavras são consideradas no cálculo da distância entre dois histogramas utilizando o resultado da equação 6.7:

$$T(BOP_x) = \sum_{k=1}^n f(p_k)_{\forall p | BOP_x(p) \notin BOP_y(p)} \quad \text{Equação 6.7}$$

onde  $T(BOP_x)$  representa o somatório das frequências das palavras  $f(p_k)$  do histograma  $BOP_x$  se, e somente se,  $\forall p | BOP_x(p) \notin BOP_y(p)$ . Para as demais palavras contidas em ambos os histogramas  $BOP_x$  e  $BOP_y$  a fórmula da distância é representada na Equação 6.8:

$$D(BOP_x, BOP_y) = \sum_{k=1}^n BOP_x(f(p_k)) - BOP_y(f(p_k))_{\forall p | BOP_x(p) = BOP_y(p)} \quad \text{Equação 6.8}$$

onde  $D(BOP_x, BOP_y)$  representa o somatório da diferença entre as frequências das palavras,  $f(p_k)$  se, e somente se,  $\forall p | BOP_x(p) = BOP_y(p)$ . Além disso, a diferença é calculada sempre da maior frequência para menor frequência para evitar valores negativos. O BOSS revolve esse problema elevando o resultado da diferença ao quadrado. Por fim, o cálculo da distância entre dois histogramas é definido pela equação 6.9:

$$DIST(BOP_x, BOP_y) = D(BOP_x, BOP_y) + T(BOP_x) \quad \text{Equação 6.9}$$

A próxima etapa consiste em definir o nível de similaridade entre o histograma não rotulado e os histogramas uBOP, nBOP ou rBOP. Basicamente, as decisões baseadas na similaridade seguem três caminhos:

- 1) Se o histograma não rotulado for similar a um rBOP, então executa a tarefa de classificação de histogramas baseado nos rBOP;
- 2) Se o histograma não rotulado for similar a um nBOP, então executa a tarefa de classificação de histogramas baseado nos nBOP. Além disso, verifica a frequência de repetições do nBOP classificado ao longo do tempo e, se necessário, transforma-o em um rBOP;
- 3) Se o histograma não rotulado for similar a um uBOP, então verifica a frequência de repetições do uBOP ao longo do tempo e, se necessário, transforma-o em nBOP por meio da aprendizagem ativa.

Os níveis de similaridade são divididos em quatro categorias:

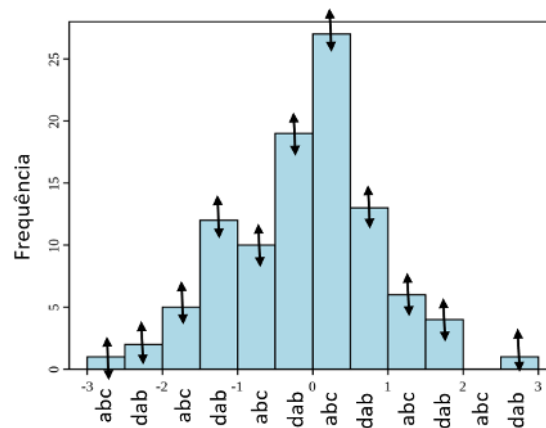
1. **Histogramas iguais:** quando a distância entre dois histogramas é igual a 0;
2. **Histogramas similares:** quando a distância entre dois histogramas está dentro de um percentual de tolerância definido como parâmetro;
3. **Histogramas similar com folga:** são histogramas similares, porém com um percentual de tolerância maior;
4. **Histogramas diferentes:** quando as palavras de dois histogramas são totalmente diferentes ou a distância entre um e outro é maior que o percentual de tolerância;

As equações 6.10 e 6.11 mostram as restrições de  $T(BOP_x)$  e  $D(BOP_x, BOP_y)$  baseado em um percentual de tolerância representado por  $\theta$ .

$$T(BOP_x) < \sum_{k=1}^n BOP_x(f(p_k)) * \theta \quad \text{Equação 6.10}$$

$$D(BOP_x, BOP_y) < \sum_{k=1}^n BOP_y(f(p_k)) * \theta \quad \text{Equação 6.11}$$

onde  $T(BOP_x)$  deve ser menor que o somatório das frequências das palavras do maior histograma  $BOP_x$  multiplicado por  $\theta$  e  $D(BOP_x, BOP_y)$  deve ser menor que o somatório das frequências das palavras do menor histograma  $BOP_y$  multiplicado por  $\theta$ . O NOHAR utiliza como valores padrões  $\theta = 0.2$  (20%) para medir histogramas similares e  $\theta = 0.3$  (30%) para medir histogramas com folga. Esses valores foram definidos com base nos experimentos descritos no Capítulo 7. A Figura 6.12 mostra um exemplo de como uma tolerância é caracterizada. Basicamente, cada histograma possui uma margem que permite a variação de frequência das palavras. Essa margem não é, necessariamente, distribuída para todas as palavras de igual maneira, ou seja, a variação pode ocorrer em uma ou um grupo de palavras.



**Figura 6.12 Exemplo de tolerância representada por folgas aceitáveis que podem variar em um histograma de referência.**

A próxima etapa consiste em descartar os histogramas diferentes e analisar somente os histogramas iguais, similares e similares com folga. O objetivo é descobrir o histograma com a menor distância ao histograma não rotulado utilizando uma adaptação do algoritmo 1-NN (*Nearest Neighbour*) usando a fórmula da equação 6.9. Desse modo, o histograma com menor distância é tratado da seguinte forma:

1. Se for um uBOP, incrementa a quantidade de repetições do histograma. Se a quantidade de repetições alcançar um limiar definido como parâmetro, então

transforma o uBOP em nBOP e atribui um rótulo por meio da interação com o usuário (aprendizagem ativa).

2. Se for um nBOP, incrementa a quantidade de repetições do histograma. Se a quantidade de repetições alcançar um limiar definido como parâmetro, então transforma o nBOP em rBOP. Classifica o histograma.
3. Se for um rBOP. Classifica o histograma.

O buffer *uBOP* possui uma quantidade limitada de histogramas desconhecidos definida como parâmetro. A medida que novos uBOP surgem, o uBOP mais antigo é descartado.

Além disso, se o histograma com menor distância for da categoria similar com folga, então um mecanismo de atualização de histogramas é acionado com o propósito de verificar a evolução de conceitos (*concept drift*). Nesse caso, o *concept drift* é caracterizado por tolerar pequenas mudanças no histograma sem gerar falsos positivos com a criação de um novo rótulo. Dessa forma, o NOHAR faz um ajuste nos histogramas uBOP, nBOP ou rBOP baseado na média da frequência das palavras iguais, conforme mostra a equação 6.12:

$$f(p_k) = \frac{BOP_x(f(p_k)) + BOP_y(f(p_k))}{2} \quad \forall p \mid BOP_x(p) = BOP_y(p) \quad \text{Equação 6.12}$$

onde  $f(p_k)$  significa uma nova frequência para a palavra  $p_k$  baseada na média das frequências dessa palavra em ambos os histogramas.

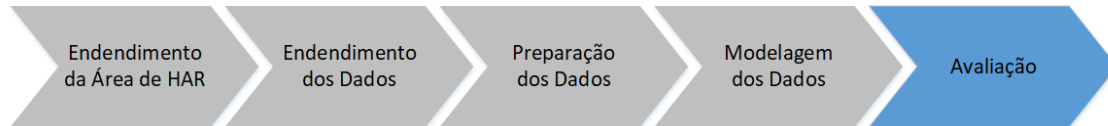
A última tarefa consiste em um mecanismo de esquecimento gradual de atividades obsoletas, ou seja, atividades que não são mais executadas pelo usuário durante um determinado período de tempo. A decisão de esquecer uma atividade é associada a pesos que refletem o período de tempo no qual o rótulo não é mais utilizado. Basicamente, a importância de um rótulo é dada pela frequência da sua utilização, isto é, quanto menor a frequência de execução de uma atividade, menor é a importância dela no modelo de classificação. A técnica utilizada para definir o peso de um rótulo é baseada na função exponencial proposta por Klinkenberg [218]. Assim, a função consiste em  $w_\gamma(H) = \exp(-\gamma t)$ , onde  $H$  é o histograma no tempo  $t$  acompanhado do fator  $\gamma$  que define a velocidade de decaimento do peso. Quanto maior o valor de  $\gamma$  menor é o peso. Pesos baixos significam histogramas de referência menos importantes e, portanto, atividades desatualizadas. Inicialmente,  $\gamma = 0$ , depois o  $\gamma$  pode ser alterado após  $H$  não ser mais classificado por um determinado período de tempo. Quando  $w_\gamma$  atingir um valor  $X$ , então o usuário é solicitado para aprovar ou não a exclusão da atividade. O tempo limite de uma atividade obsoleta é definida como parâmetro pelo NOHAR.

## 6.4. Considerações Finais

Este capítulo apresentou a principal contribuição desta tese de doutorado. Aqui foram descritos o *framework* DISTAR projetado para lidar com futuros algoritmos simbólicos adaptativos e o algoritmo NOHAR que é uma instância do *framework* DISTAR. Mais informações relacionadas operação do algoritmo proposto no contexto de HAR será apresentada no Capítulo 7 no decorrer dos experimentos. Além disso, diversas comparações com o estado da arte em termos de precisão, consumo de memória e tempo de execução foram feitas.

# CAPÍTULO 7

## EXPERIMENTOS E RESULTADOS



Este capítulo apresenta os experimentos e resultados obtidos pelo método NOHAR, bem como as avaliações dos métodos e técnicas discutidos neste trabalho.

### 7.1 Protocolo Experimental

Esta seção apresenta uma visão geral de como os experimentos foram planejados, organizados e executados no contexto de HAR em smartphones. Os experimentos foram agrupados em três cenários. Para cada cenário são apresentados os *baselines* e os parâmetros utilizados pelos algoritmos avaliados. Os cenários dos experimentos são:

1. Avaliação da metodologia tradicional de HAR usando os algoritmos de classificação *offline* rasos e profundos. Os experimentos avaliam o impacto causado pelos métodos e técnicas utilizados em cada etapa da metodologia, como segmentação, extração e seleção de características e fusão de dados. Nesse sentido, os resultados são focados na acurácia dos modelos de classificação para cada cenário avaliado. O objetivo desses experimentos é obter resultados para facilitar a comparação das soluções tradicionais com as soluções baseadas nos algoritmos de representação simbólica.
2. Avaliação dos algoritmos de classificação simbólicos SAX-VSM, BOSS-VS e WEASEL no contexto de HAR para smartphones. A avaliação consiste em uma análise comparativa entre os algoritmos de representação simbólica e os demais algoritmos de classificação rasos Árvore de Decisão, *Naive Bayes*, KNN e SVN comumente utilizado pela literatura tradicional de HAR. O objetivo é mostrar os benefícios do uso dos algoritmos de representação simbólica em termos de acurácia dos modelos de classificação, consumo de memória e tempo de processamento.
3. Avaliação do método proposto por esta tese, o NOHAR, juntamente com uma análise comparativa com outras abordagens *online* disponíveis na literatura como *Hoeffding Tree*, *Hoeffding Adaptive Tree* e *AnyNovel* descritas na Seção 5.4.

Os experimentos foram conduzidos utilizando uma ferramenta de avaliação de sistemas HAR (BHAR – *Benchmarking HAR*) desenvolvida durante o período desta tese de doutorado. A ferramenta BHAR consiste em uma ferramenta de *benchmarking* que contém a implementação dos métodos comumente utilizados na literatura para reconhecer as atividades humanas no contexto da análise de dados de sensores inerciais. A ferramenta BHAR foi criada com intuito de facilitar a avaliação e a comparação de diferentes abordagens utilizadas na literatura. Além disso, as implementações referentes a cada abordagem podem ser facilmente reaproveitadas, via uma API (*Application Programming Language*) fornecida pela ferramenta para gerar aplicações específicas capazes de executar nos smartphones. A Figura 7.1 mostra a

tela principal da ferramenta, mais detalhes sobre essa ferramenta são apresentados no Apêndice D.



Figura 7.1 Tela principal da ferramenta BHAR. Fonte: autor.

### 7.1.1 Métricas

As métricas utilizadas nos experimentos são a acurácia (ver Seção 3.6.3), o tempo de processamento dos algoritmos em milissegundos e o consumo de memória em Bytes. A acurácia foi escolhida devido ser a principal métrica de avaliação dos modelos de classificação no estado da arte. Além disso, duas das bases de dados utilizadas nos experimentos são balanceadas, o que permite a consistência dos resultados da acurácia. Os experimentos foram executados utilizando a estratégia *Cross-validation 10-folds*. Cada *fold* contém dados aleatórios pertencentes a qualquer usuário da base de dados. Essa estratégia também foi escolhida para facilitar a comparação dos nossos resultados com os resultados alcançados na literatura.

### 7.1.2 Bases de dados utilizados nos experimentos

Os experimentos desta tese foram baseados em três bases de dados comumente utilizadas na literatura para avaliar as soluções de HAR: UCI-HAR, Shoaib e WISDM. Esta seção apresenta apenas um resumo para cada uma delas, mais detalhes estão descritos no Apêndice A.

A base de dados UCI-HAR<sup>3</sup> [219] foi construída com dados coletados de 30 pessoas com idades que variam entre 19 e 48 anos. Cada pessoa executou 6 atividades físicas como andar, sentar, ficar em pé, deitar, subir e descer escadas. Os dados foram coletados de um celular *Samsung Galaxy S2* utilizando os sensores acelerômetro e giroscópio a uma frequência de 50 Hz. A coleta foi realizada com o celular localizado na cintura dos usuários. Todas as

<sup>3</sup> <https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

etapas de coleta de dados foram gravadas<sup>4</sup> e os dados foram rotulados manualmente. A UCI-HAR possui 1.311.439 amostras.

A base de dados Shoaib [91] foi construída com dados de 10 usuários do sexo masculino com idades variando de 25 e 30 anos. Cada usuário executou 8 atividades no período de 3 a 4 minutos para cada uma. Tais atividades incluem andar, correr, sentar, ficar de pé, correr lentamente, andar de bicicleta, subir escadas e descer escadas. Os dados foram coletados de um celular *Samsung Galaxy S2* (i9100) utilizando os sensores acelerômetro, acelerômetro linear, giroscópio e magnetômetro a uma frequência de 50 Hz. Os usuários foram equipados com cinco smartphones localizados em cinco posições do corpo incluindo bolso direito e esquerdo da calça, cintura, pulso e antebraço. A Shoaib possui 629.977 amostras.

A base de dados *Wireless Sensor Data Mining* (WISDM) [81] foi construída com dados de 36 usuários. Cada usuário executou 6 atividades como andar, sentar, ficar em pé, deitar, subir e descer escadas. Os dados foram coletados de um smartphone com Android (*Nexus One*, *HTC Hero*) a uma frequência de 20Hz. A coleta foi realizada com o celular localizado no bolso frontal da calça do usuário. A WISDM possui 1.098.830 amostras.

## 7.2. Avaliação dos Algoritmos Rasos e Profundos

Esta seção mostra o resumo da avaliação do estado da arte de HAR na perspectiva dos algoritmos *offline* com destaque para a metodologia tradicional baseada nas etapas de segmentação, extração e seleção de características e fusão de dados. Como essa avaliação não é o foco desta tese, os detalhes dos experimentos estão descritos no Apêndice C. Em resumo, esta seção reúne as principais conclusões dos resultados obtidos pelos experimentos realizados nesta tese confirmados pelos experimentos reportados na literatura:

- 1) Em geral, o acelerômetro é o sensor predominante no processo de reconhecimento de atividades físicas dos usuários.
- 2) Os dados extraídos do giroscópio complementam os dados do acelerômetro e ambos geram modelos de classificação melhores com um aumento médio 2% na acurácia. Esses resultados também foram observados em [91] [27] [28].
- 3) Os dados do acelerômetro sem as informações de gravidade (acelerômetro linear) geram modelos de classificação com menor acurácia comparado aos dados do acelerômetro com informação de gravidade. Esses resultados também foram observados por [92].
- 4) O tamanho ideal para as janelas de tempo fixas varia em torno de 2 a 5 segundos considerando uma frequência de 20Hz a 50 Hz. Esses resultados também foram observados em [87] [92] [27].
- 5) A cintura e o bolso da calça é a melhor posição para reconhecer atividades físicas simples como andar e correr, visto que a origem dos movimentos do corpo humano ocorre nessas regiões [80].
- 6) As características do domínio do tempo, especialmente as: média, variância, desvio padrão, *Root Mean Square*, mínimo, máximo, amplitude e correlação, geram modelos com maior acurácia comparado as características do domínio de frequência. Além disso, as características de tempo são mais baratas e consomem

<sup>4</sup> [https://www.youtube.com/watch?v=XOEN9W05\\_4A](https://www.youtube.com/watch?v=XOEN9W05_4A)

menos energia da bateria em relação as características do domínio de frequência [86].

- 7) As características derivadas do Wavelet são melhores do que as características derivadas do FFT.
- 8) As características independentes de orientação derivadas da magnitude e componentes verticais e horizontais não são suficientes para representar bem as atividades físicas.
- 9) Os classificadores rasos da família da árvore decisão obtiveram as melhores taxas de acurácia no contexto dos sensores inerciais.
- 10) As redes neurais profundas superam os algoritmos rasos. No entanto, exigem uma alta carga de processamento na etapa de treinamento dos modelos de classificação.

### 7.3. Avaliação dos Algoritmos de Representação Simbólica

Esta seção apresenta uma análise comparativa entre os modelos de classificação gerados pelos algoritmos de representação simbólica SAX-VSM, BOSS-VS e WEASEL e os modelos gerados pelos algoritmos rasos Árvore de Decisão, *Naive Bayes*, KNN e SVM. Os resultados são apresentados conforme as métricas de avaliação. Os primeiros resultados apresentam a avaliação da acurácia dos modelos de classificação. Um segundo conjunto de experimentos é realizado para avaliar o tempo de processamento nas etapas de extração de características e treinamento dos modelos. Por fim, os resultados referentes a avaliação do consumo de espaço (memória) de cada estratégia são apresentados.

#### 7.3.1 Baselines

Os *baselines* utilizados na comparação com as técnicas de representação simbólica são os algoritmos rasos Árvore de Decisão, *Naive Bayes*, KNN (*k-Nearest Neighbors*) e o SVM (*Support Vector Machine*) comumente utilizados na literatura de HAR [123] [119] [54] [120]. Os algoritmos profundos não foram inclusos nesses experimentos devido ao alto processamento exigindo na etapa de treinamento do modelo de classificação e pela dificuldade de implementar uma solução completamente dependente do smartphone. Os algoritmos rasos foram treinados com base nas características do domínio do tempo e da frequência listadas na Tabela 7.1, sendo que as características de frequência são extraídas com base nos resultados da FFT e *Wavelet*. Os detalhes sobre cada uma dessas características estão descritos na Seção 3.4.

**Tabela 7.1** Lista das características do domínio da frequência e do tempo.

Domain	Features
Time	min, max, amplitude, amplitude peak, sum, absolute sum, Euclidian norm, mean, absolute mean, mean square, mean absolute deviation, sum square error, variance, standard deviation, Pearson coefficient, zero crossing rate, correlation, cross-correlation, auto-correlation, skewness, kurtosis, area, absolute area, signal magnitude mean, absolute signal magnitude mean, magnitude difference function.
Frequency	Energy, energy normalized, power, centroid, entropy, DC component, peak, coefficient sum.



### 7.3.2 Configurações e Parâmetros

Esta seção apresenta as configurações e os parâmetros utilizados pelos *baselines* e pelos algoritmos de representação simbólica. Para simplificar a apresentação dos resultados, os experimentos foram realizados somente com os dados do sensor inercial acelerômetro que é considerado na literatura como suficiente para representar as atividades físicas dos usuários.

As configurações e parâmetros utilizados para os *baselines* são baseadas nos melhores resultados obtidos nos experimentos descritos no Apêndice C, onde:

- **Tamanho da janela de tempo:** 5 segundos.
- **Taxa de sobreposição:** 50%.
- **Características:** domínio do tempo, domínio da frequência derivada do FFT e domínio da frequência derivada do Wavelet.

As configurações e parâmetros utilizados para os algoritmos de representação simbólica incluem: 1) uso das estratégias de manipulação de séries multidimensionais; 2) tamanho da janela de tempo; 3) tamanho da palavra; e 4) tamanho do alfabeto. Alguns parâmetros foram definidos com base em sugestões da literatura e outros parâmetros com base em experimentos. O estudo dos melhores parâmetros foi realizado usando a base de dados UCI-HAR e o resultado foi replicado para as demais bases de dados. Assim, as configurações definidas foram:

- **Tamanho da janela de tempo:** os algoritmos foram avaliados com diferentes tamanhos de janelas de tempo. Os tamanhos variaram entre 20 e 250 amostras. Para cada base de dados, o melhor tamanho da janela de tempo encontrado foi igual à quantidade de amostras contidas em 1 segundo, ou seja, valor equivalente a taxa de frequência da coleta da base de dados. Por exemplo, para as bases de dados UCI-HAR e SHOAIB o melhor tamanho de janela encontrado foi de 50 amostras e, para a base de dados WISDM, o melhor tamanho de janela foi de 20 amostras.
- **Tamanho da palavra:** os algoritmos foram avaliados com diferentes tamanhos de palavras que variaram entre 4 e 8 letras. Esses tamanhos foram escolhidos devido serem comumente utilizados pelos autores do SAX e SFA. Nesse caso, foi observado que quanto maior o tamanho da palavra, melhor é a acurácia dos modelos de classificação. Isso ocorre porque as palavras maiores tendem a se aproximar mais da subsequência original. Portanto, o tamanho da palavra que ficou definido foi de 8 letras.
- **Tamanho do alfabeto:** O tamanho do alfabeto foi fixado em 4 caracteres, conforme recomendado pelos autores do algoritmo SAX [38] e SFA [39].

### 7.3.3 Resultados

#### 7.3.3.1 Acurácia dos Modelos de Classificação

Esta seção apresenta os resultados referentes à acurácia dos modelos de classificação. A Tabela 7.2 apresenta um resumo dos resultados referentes aos modelos de classificação gerados pelos algoritmos rasos. Os resultados foram agrupados por domínio de características. O primeiro grupo apresenta os resultados dos modelos gerados com as características do domínio do tempo. O segundo e o terceiro grupo correspondem aos resultados dos modelos gerados com as características do domínio de frequência baseadas no FFT e *Wavelet*.

**Tabela 7.2 Resultados referentes a acurácia dos modelos de classificação geradas pelos algoritmos rasos combinados com as características do domínio do tempo e da frequência.**

Features	Algorithm	UCI-HAR	SHOAIB	WISDM
Time Features	Decision Tree	84.64	94.56	88.38
	Naive Bayes	57.36	73.74	67.34
	SVM	81.78	92.04	83
	KNN	87.83	96.76	87.72
FFT	Decision Tree	69.48	76.02	68.05
	Naive Bayes	27.81	57.39	35.03
	SVM	60.12	70.76	63.51
	KNN	65.84	74.08	65.94
Wavelet	Decision Tree	74.96	79.22	77.85
	Naive Bayes	25.78	69	45.95
	SVM	60.17	68.22	72.58
	KNN	61.03	63.72	73.53

Como pode ser observado, os resultados mostram que os classificadores baseados nas características do domínio do tempo são superiores, em termos de acurácia, aos classificadores baseados nas características do domínio da frequência. Nesse caso, o destaque deve ser dado aos classificadores *Árvore de Decisão* e *KNN*, pois ambos possuem uma taxa de acurácia média de 89.19% e 90.77%, respectivamente, considerando todas as bases de dados avaliadas. Por outro lado, as taxas de acurácias dos mesmos classificadores baseados nas características do domínio de frequência (*FFT* e *Wavelet*) foram reduzidas, em média, 22.31%. Além disso, dentre todos os classificadores, o de pior desempenho foi o *Naive Bayes* com uma acurácia média de 66.15% para as características do domínio do tempo.

**Tabela 7.3 Resultados referentes a acurácia dos modelos de classificação geradas pelos algoritmos de representação simbólica utilizando as estratégias de fusão de coordenadas por empilhamento, magnitude e PCA.**

Features	Algorithm	UCI-HAR	SHOAIB	WISDM
Axis Fusion	SAX-VSM	95.7	91.3	75.1
	BOSS-VS	98.5	99.9	94.4
	WEASEL	52.9	48.4	59.9
Magnitude	SAX-VSM	87.7	81.7	60.8
	BOSS-VS	98.9	99.6	96
	WEASEL	36.6	54.09	57.9
PCA	SAX-VSM	69.6	80.7	36.5
	BOSS-VS	96.9	99.7	97.2
	WEASEL	41	43.6	62.4

A Tabela 7.3 mostra os resultados referentes aos algoritmos de representação simbólica. Os resultados foram agrupados pelas estratégias de fusão de coordenadas descritas na Seção 4.4. O primeiro grupo apresenta os resultados dos modelos de classificação gerados com a estratégia de empilhamento sincronizado das palavras de cada coordenada. O segundo e terceiro grupo correspondem aos resultados dos modelos gerados com as estratégias magnitude e PCA, respectivamente.

Como pode ser observado, o algoritmo de representação simbólica BOSS-VS apresenta o melhor resultado com uma acurácia média de 97.9%. Esse resultado ocorre por dois motivos: o primeiro é porque o SFA tem uma alta capacidade de aproximação da série temporal original devido a etapa de quantização ser baseada nas posições variáveis dos *breakpoints* definidos pelos coeficientes do FFT. O segundo motivo é que para esses experimentos foi utilizada a estratégia de avaliação baseada na validação cruzada onde cada partição contém dados aleatórios de toda a base de dados.

Os resultados da Tabela 7.2 e da Tabela 7.3 mostram que os algoritmos de representação simbólica obtiveram as melhores acurácias, quando comparadas aos algoritmos rasos combinados com as características do domínio do tempo. Portanto, isso mostra a eficiência desses algoritmos no contexto de HAR, em termos de precisão dos modelos de classificação. As próximas seções mostram a eficiência desses algoritmos em termos de tempo de processamento e consumo de memória.

### 7.3.3.2 Tempo de Processamento

Esta seção apresenta os resultados referentes ao tempo de processamento das etapas de extração de características e treinamento dos modelos de classificação no contexto dos algoritmos rasos e de representação simbólica.

A Tabela 7.4 mostra o tempo de processamento da etapa de extração de características tanto para os domínios do tempo e frequência quanto para as características do domínio discreto extraídas pelos algoritmos SAX, SFA e SFA-W. Considerando apenas os resultados da base de dados WISDM com 1.098.830 de amostras, o tempo médio para extrair as características dos domínios do tempo e da frequência é de 12461.33 milissegundos. Enquanto que o tempo médio para discretizar os dados com os algoritmos avaliados são de 1892.66 milissegundos. Isso representa uma redução de 84.81% do processo de extração de características do domínio discreto em relação aos demais domínios.

**Tabela 7.4** Lista de tempo de processamento na etapa de extração de características.

Feature Extraction Time (ms)			
Features	UCI-HAR	SHOAIB	WISDM
Time Features	14458	6434	11240
Frequency Features (FFT)	17008	8227	17750
Frequency Features (Wavelet)	13519	5387	8394
SAX	3121	1706	2636
SFA	559	301	418
SFA-W	4230	2858	2624

Em especial, o SFA é o algoritmo mais rápido (com 418 milissegundos) devido ser otimizado para tratar as palavras como um conjunto de *bits*. Por exemplo, um alfabeto de tamanho 4 é representado por apenas 2 bits, ou seja, se as letras do alfabeto forem `a`, `b`, `c` e `d`, então a representação de cada letra será `a=00`, `b=10`, `c=11`, `d=01`. Dessa forma, o ganho de tempo no processamento alcança 96.28%. Por outro lado, o algoritmo SAX manipula as palavras sem nenhuma otimização e, mesmo assim, o tempo de processamento é de 2636 milissegundos. O tempo de processamento do SFA-W é equivalente ao do SAX devido aos processamentos de seleção de palavras importantes com a aplicação do ANOVA Test e ganho de informação.

A Tabela 7.5 mostra o tempo de processamento durante o treino dos modelos de classificação pelos classificadores rasos e os de representação simbólica. Um olhar atento à Tabela 7.5 mostra que a média de tempo de processamento dos classificadores rasos *Naive Bayes* e KNN é similar ao classificador simbólico SAX-VSM. Da mesma forma, ocorre com os algoritmos BOSS-VS, Árvore de Decisão e SVM. Por outro lado, o algoritmo Weasel é o classificador com o maior tempo de processamento com uma média de 18961 ms. O motivo pelo qual o Weasel é um classificador lento é porque os histogramas são compostos por unigramas e bigramas de palavras, além de considerar diferentes tamanhos de palavras e janelas de tempo (ver Seção 4.4.2.2).

**Tabela 7.5 Lista de tempo de processamento na etapa de treinamento do modelo de classificação.**

Algorithm	Train Time (ms)		
	UCI-HAR	SHOAIB	WISDM
KNN	51	45	49
Naive Bayes	151	66	125
Decision Tree	2534	925	2507
SVM	4073	1277	9233
SAX-VSM	76	97	39
BOSS-VS	8895	2014	1976
WEASEL	31036	7531	18318

Por fim, com base nos resultados da Tabela 7.4 e Tabela 7.5 é possível concluir que o SAX-VSM é o algoritmo com menor tempo de processamento com uma duração média de 2558 milissegundos, considerando a soma do tempo de processamento das etapas de extração de características e treinamento do modelo do modelo de classificação. O BOSS-VS é o segundo algoritmo mais rápido com uma duração média de 4721 milissegundos nas duas etapas. Além disso, é importante destacar que as estratégias de fusão de coordenada baseadas na magnitude e PCA são as que exigem menor custo de processamento dos dados comparado a estratégia de empilhamento de coordenadas. Mais especificamente, a estratégia baseada na magnitude do sinal é a que consome menos recursos computacionais.

### 7.3.3.3 Consumo de Memória

Esta seção apresenta os resultados referentes ao consumo de memória antes e depois do processo de extração de características para as bases de dados UCI-HAR, WISDM e SHOAIB. Os resultados estão descritos na Tabela 7.6 em forma da unidade de medida *Bytes*. A estratégia utilizada para coletar esses dados foi a seguinte: 1) calcular a quantidade de *Bytes* que os dados brutos ocupam na memória; 2) aplicar o processo de extração de características; e 3) calcular a quantidade de *Bytes* que os dados resultantes do processamento ocupam na memória. Para as características do domínio da frequência foram escolhidas somente as características derivadas do FFT, visto que as características derivadas do *Wavelet* possuem resultados semelhantes. Para as características do domínio discreto foi escolhido o algoritmo SAX como algoritmo de discretização com o tamanho de palavra 8 e alfabeto 4.

Como pode ser observado, as características do domínio do tempo conseguem reduzir, em média, 73.90% dos dados originais para todas as bases de dados. Por exemplo, os dados da UCI-HAR foram reduzidos de 43 milhões para 9 milhões de Bytes aproximadamente. Para as características do domínio da frequência o valor da redução dos dados subiu para 92.66%. O

motivo é que a quantidade de característica do domínio de frequência é 68.96% menor comparado às características do domínio de tempo, ou seja, os experimentos foram executados com 29 características de tempo e 9 características de frequência, conforme mostra a Tabela 7.1. Por outro lado, as características do domínio discreto reduzem o espaço de memória em média 94.48%. Um ganho de 1.92% em relação às características do domínio de frequência. Isso mostra a capacidade de compressão de dados dos algoritmos de representação simbólica mantendo a precisão dos modelos de classificação que utilizam esses algoritmos.

**Tabela 7.6 Consumo de memória em Bytes dos dados antes e depois do processo de extração de características.**

Dataset	All	Time Features	Frequency Features (FFT)	Discrete Features (SAX)
UCI-HAR	43,409,405	9,014,107	2,503,597	1,779,486
WISDM	22,035,537	7,019,442	1,967,915	866,972
SHOAIB	16,175,747	4,151,907	1,182,459	1,377,616

## 7.4. Avaliação do NOHAR

Esta seção apresenta detalhes de como os algoritmos *online* baseados em *data streaming* são avaliados. Os algoritmos *online* são avaliados de forma diferente dos algoritmos *offline*, visto que não existe mais uma divisão da coleção de dados entre treino e teste. No contexto *offline*, a geração de um modelo de classificação depende do processamento integral da base de dados de treino que é carregada completamente na memória antes do processamento. Em seguida, as instâncias disponíveis na base de dados de teste são utilizadas para avaliar o modelo de classificação.

No contexto *online*, a estratégia de avaliação muda porque a base de dados cresce de forma ilimitada e constante ao longo do tempo. Isso torna o carregamento do histórico de dados na memória inviável. Nesse caso, a estratégia adotada é baseada na análise dos dados contidos em um *buffer*. Desse modo, a estratégia de avaliação utilizada nesta tese de doutorado é a Prequential (*predictive sequential*) [36], ou seja, as amostras utilizadas para treinar e atualizar o modelo de classificação são as mesmas utilizadas na avaliação. Essa estratégia é baseada na premissa das previsões de probabilidades sequenciais para observações futuras, em vez de observar o passado com a estratégia de avaliação Holdout.

As métricas utilizadas para avaliar os modelos de classificação *online* são as mesmas utilizadas pelos algoritmos *offline*. A diferença é que as taxas de acurácia e tempo de processamento são representadas pelas médias dos valores ao longo do tempo [36] [212]. Os experimentos foram divididos em três cenários com base nos *baselines* descritos na Seção 5.4:

1. **NOHAR**: avaliação do algoritmo NOHAR com o propósito de mostrar um panorama detalhado do algoritmo no contexto de HAR.
2. **NOHAR x AnyNovel**: análise comparativa entre os algoritmos NOHAR e o AnyNovel, incluindo o desempenho dos modelos de classificação, tempo de processamento e uso do módulo de aprendizagem ativa. O AnyNovel é o principal *baseline* deste trabalho por ter sido criado especificamente para a área de HAR baseada em smartphones com sensores inerciais.
3. **NOHAR x Hoeffding Tree x Hoeffding Adaptive Tree**: análise comparativa entre os principais algoritmos *online* da literatura considerados o estado da arte. A análise

inclui a análise de desempenho dos modelos de classificação e o tempo de processamentos dos algoritmos.

#### 7.4.1 Baselines

Os algoritmos utilizados como *baselines* para avaliar o NOHAR estão descritos na Seção 5.4. Basicamente, os algoritmos selecionados são baseados nas estratégias de aprendizagem estacionária, evolução dos dados e detecção de novidades. Nessa perspectiva, os algoritmos selecionados que correspondem a essas categorias são *Hoeffding Tree* (HT) [213], *Hoeffding Adaptive Tree* (HAT) [36] e AnyNovel [26], respectivamente.

#### 7.4.2 Configurações e Parâmetros

##### 7.4.2.1 Parâmetros NOHAR

Esta seção descreve a configuração e os parâmetros utilizados pelo algoritmo NOHAR. Tais parâmetros foram definidos com base no conhecimento do domínio de HAR baseada em sensores inerciais, em sugestões fornecidas pela literatura e em experimentos realizados neste trabalho. É importante ressaltar que esses parâmetros foram utilizados para todos os demais experimentos independentes da base de dados utilizadas. Os parâmetros definidos são os seguintes:

- **Tamanho da janela de tempo:** no contexto de HAR baseado em sensores inerciais, a coleta de dados baseada em 20 amostras por segundo (20 Hz) é suficiente para representar o movimento humano [27] [87]. Por este motivo, o valor padrão definido para o NOHAR é de 20 amostras independente da frequência da coleta de dados.
- **Tamanho da palavra:** por padrão o NOHAR utiliza palavras de tamanho 4, visto que o problema da aproximação da palavra com a sequência original é minimizado com o melhoramento da qualidade do sinal por meio da aplicação do filtro *Exponential Smoothing* descrito na seção Seção 6.3.1.
- **Tamanho do alfabeto:** O tamanho do alfabeto foi fixado em 4 letras, conforme recomendado pelos autores do algoritmo SAX [38] e SFA [39].
- **Offset:** o *offset* significa a quantidade de amostras da sequência que são puladas antes de gerar uma nova palavra. No NOHAR, o *offset* é igual ao resultado da razão entre os tamanhos da janela de tempo e da palavra. Por exemplo, para os tamanhos das janelas de tempo igual a 20 e da palavra igual a 4, o *offset* é igual a 5. Isso significa que duas subsequências subjacentes não possuem sobreposição entre si, tornando as letras independentes umas das outras.
- **Tamanho *buffer* uBOP:** quando um novo histograma é desconhecido, ele é armazenado no *buffer* uBOP. Por padrão, o NOHAR mantém armazenado na memória 100 histogramas. Quando o *buffer* está completo, sempre o histograma mais antigo é descartado da memória.
- **Tamanho *buffer* nBOP:** por padrão, quando um uBOP aparece 3 vezes ao longo tempo, ele é transformado em um nBOP. O *buffer* nBOP tem a capacidade de armazenar 50 histogramas. Quando o *buffer* está completo, sempre o histograma mais antigo é descartado da memória.
- **Histogramas similares:** o NOHAR considera dois histogramas similares quando o percentual de tolerância da distância entre eles não excede 20%. Esse valor foi definido por meio de experimentos.

- **Histogramas similares com folga:** O NOHAR considera dois histogramas similares com folga quando o percentual de tolerância da distância entre eles não excede 30%. Esse valor foi definido por meio de experimentos.
- **Tamanho do BOP:** no contexto de HAR baseado em smartphones com sensores inerciais, 5 segundos de dados é suficiente para reconhecer uma atividade humana [92]. Portanto, 5 segundos é valor padrão que define a quantidade de dados que um histograma BOP do NOHAR representa no contexto dos sensores inerciais.

#### 7.4.2.2 Parâmetros AnyNovel

Esta seção descreve as seguintes configurações e os parâmetros utilizados pelo algoritmo AnyNovel:

- **Tamanho da janela de tempo:** 5 segundos.
- **Faixa de tolerância do cluster (DRAB):** Esse parâmetro consiste em uma região em torno do cluster responsável por monitorar a evolução dos conceitos (*concept drifting*). Nesse caso, o parâmetro utilizado pelo autor do AnyNovel é 10%, ou seja, é aceitável que os dados contidos no *buffer* estejam, no máximo, 10% fora do cluster original.
- **Tamanho estável da novidade:** número mínimo de pontos requeridos para considerar um conjunto de dados como sendo um novo conceito. O valor desse parâmetro foi definido em 200 amostras que é o dobro do tamanho da janela de tempo.

O algoritmo AnyNovel possui duas fases: *offline* e *online*. A fase *offline* consiste em um pré treino do modelo de classificação com uma base de dados de treino inicial e a fase *online* consiste na adaptação do modelo de classificação ao longo do tempo conforme novas entradas são analisadas. Para facilitar a apresentação dos resultados, os dados referentes a atividade andar, escolhida aleatoriamente, foram removidos da base de dados de treino com o propósito de verificar a capacidade do AnyNovel de identificar essa atividade como sendo uma novidade na base de dados de teste. Abdallah et al. [26] utilizaram a mesma estratégia na apresentação de seus resultados.

#### 7.4.2.3 Parâmetros do Hoeffding Tree e Hoeffding Adaptive Tree

Os parâmetros utilizados pelas árvores de decisões dos algoritmos HT e HAT são:

- **Tamanho da janela de tempo:** esse parâmetro foi definido com base nos mesmos motivos explicados na Seção 7.4.2.1, ou seja, 100 amostras por *chunk* ou janela de tempo.
- **Grace Period:** esse parâmetro consiste em definir um intervalo de amostras para que a árvore de decisão possa recalcular o ganho de informação com o propósito de decidir se a árvore de decisão deve ser alterada ou não. No contexto de HAR, foi definido que esse cálculo deve ser feito após a análise de cada *chunk*, ou seja, a cada 100 amostras.

### 7.4.3 Resultados

#### 7.4.3.1 Acurácia do Algoritmo NOHAR

Esta seção apresenta os resultados dos experimentos executados somente com o algoritmo NOHAR. O objetivo é mostrar o comportamento e o desempenho do NOHAR de

forma detalhada antes de apresentar as comparações com os *baselines*. Para facilitar a apresentação dos resultados, os experimentos foram realizados usando somente a base de dados SHOIAB. Em seguida, as demais seções apresentam uma comparação detalhada do NOHAR com os *baselines* incluindo estudos com as demais bases de dados WISDM e UCI-HAR.

A base de dados SHOAI B contém dados de 10 usuários e o objetivo é mostrar a capacidade do NOHAR em reconhecer as atividades de cada um deles. Além da discussão dos erros e acertos do modelo de classificação, também são analisados a quantidade de dados e histogramas necessários para que os padrões de cada atividade sejam aprendidos e mapeados pelo modelo de classificação. A Tabela 7.7 mostra um resumo dos resultados representados pelas taxas de acurácia separadas por usuário e atividade.

**Tabela 7.7 Taxas de acurácia do algoritmo NOHAR para cada usuário da base de dados Shoaib.**

User	Accuracy	Walking	Standing	Jogging	Sitting	Biking	Upstairs	Downstairs
S1	99.60	100.00	100.00	-	100.00	100.00	80.00	-
S2	99.26	100.00	100.00	100.00	100.00	100.00	90.00	-
S3	97.46	100.00	100.00	-	100.00	100.00	91.17	-
S4	93.12	100.00	100.00	-	100.00	100.00	28.57	-
S5	88.53	100.00	100.00	100.00	84.14	96.07	52.94	25.00
S6	98.38	100.00	100.00	-	100.00	100.00	50.00	-
S7	96.41	100.00	100.00	100.00	97.59	98.18	100.00	36.36
S8	98.58	-	100.00	-	100.00	94.64	100.00	-
S9	84.90	100.00	100.00	100.00	58.74	100.00	92.30	-
S10	92.30	100.00	100.00	100.00	100.00	100.00	77.77	10.00

Como pode ser observado, as atividades *Upstairs* e *Downstairs* são as atividades que o algoritmo NOHAR possui maior dificuldade em aprender. A matriz de confusão apresentada na Tabela 7.8 mostra que a atividade *Upstairs* é bastante confundida com a atividade *Walking*, enquanto que a atividade *Downstairs* é bastante confundida com as atividades *Upstairs* e *Walking*. A confusão entre essas atividades também é vista nos demais algoritmos de classificação AnyNovel (ver Seção 7.4.4) e *Hoeffding Tree* (ver Seção 7.4.5). O problema dessa confusão é que os sinais das atividades *Upstairs* e *Downstairs* são similares entre si e ao sinal da atividade *Walking*. Para resolver esse problema, ambas as atividades podem ser tratadas por meio de um reconhecimento hierárquico. Por exemplo, em vez de reconhecer essas atividades de forma separada, poderia reconhecer uma atividade como, por exemplo, *Cimbstairs* e, em seguida, reconhecer se o usuário está subindo ou descendo as escadas.

Por outro lado, o modelo de classificação gerado pelo NOHAR é capaz de aprender muito bem as demais atividades com uma taxa acurácia de 100% para a maioria dos usuários. Alguns usuários obtiveram algumas exceções para as atividades *Sitting* e *Biking* que, de acordo com a matriz de confusão da Tabela 7.8, se confundiram entre si e com a atividade *Standing* para os dados dos usuários S5, S7 e S9. Em média, a taxa de acurácia geral para todos os usuários foi de 94.85%.

A Tabela 7.7 também apresenta algumas lacunas referentes às ausências das taxas de acurácia para algumas atividades como, por exemplo, as lacunas das atividades *Jogging* e *Downstairs* do usuário S1. Isso significa que essas atividades não foram aprendidas pelo algoritmo NOHAR, ou seja, os dados referentes às atividades *Jogging* e *Downstairs* do usuário S1 não foram suficientes para o algoritmo NOHAR encontrar um padrão. Na prática, os



histogramas gerados para ambas atividades permaneceram desconhecidos (uBOP) ao longo da análise de dados, ou seja, em nenhum momento foram transformados em nBOP (*Novel BOP*) ou rBOP (*Reference BOP*). Por outro lado, os dados do usuário S2 foram suficientes para que a atividade *Jogging* fosse aprendida pelo NOHAR, enquanto que os dados da atividade *Downstairs* continuou gerando histogramas desconhecidos. A mesma intuição pode ser refletida para as demais lacunas encontradas para outros usuários e atividades.

Tabela 7.8 Matriz de confusão do algoritmo NOHAR.

	Walking	Standing	Jogging	Sitting	Biking	Upstairs	Downstairs	Unknown
Walking	254	0	0	0	1	31	0	336
Standing	0	832	0	7	16	0	0	23
Jogging	1	0	20	0	0	1	0	702
Sitting	0	14	0	856	5	0	0	22
Biking	0	3	0	0	619	1	0	133
Upstairs	64	0	0	0	4	283	0	325
Downstairs	17	0	0	0	5	34	12	697

Outro fator importante na matriz de confusão é quantidade de histogramas desconhecidos. Como pode ser observado, as atividades *Jogging* e *Downstairs* foram as atividades que obtiveram a maior quantidade de histogramas desconhecidos e as atividades *Sitting* e *Standing* obtiveram a menor quantidade de histogramas desconhecidos. Isso mostra, novamente, o nível de dificuldade que o algoritmo NOHAR tem em aprender atividades onde o sinal do sensor inercial é aleatório ou sem padrão.

Tabela 7.9 Número de histogramas desconhecidos, de treino e teste gerados para a base de dados SHOAI B.

	Walking	Standing	Jogging	Sitting	Biking	Upstairs	Downstairs
Unknown	336	23	702	22	133	325	697
Train	32	6	8	4	19	64	9
Test	254	849	14	871	604	333	68

A Tabela 7.9 mostra uma análise quantitativa do número de histogramas gerados na base de dados SHOAI B. Ao todo foram gerados 5373 histogramas, onde apenas 142 histogramas foram utilizados para treino e 2993 histogramas foram utilizados para teste. Os restantes dos histogramas permaneceram sendo desconhecidos ao longo da análise dos dados. Além disso, ao final do treinamento, o modelo de classificação conteve 46 histogramas de referência e 50 histogramas de novidades.

### Análises ao longo do tempo

A Figura 7.2 mostra uma visão geral da acurácia do NOHAR ao longo do tempo para todos os usuários (linha superior), incluindo também os momentos em que o NOHAR solicita o rótulo por meio de requisições ao módulo de aprendizagem ativa (barras pontilhadas). Para fins de comparação com os *baselines* nas próximas seções, nós convencionamos que as requisições para a atividade andar, pelo NOHAR, são identificadas como sendo as barras pontilhadas que atingem o valor 75. Os restantes das requisições são representadas pelas barras que atingem o valor 60.

Em relação a acurácia, o NOHAR inicia com 100% porque, no início, o modelo de classificação detecta e aprende o padrão de apenas uma atividade (ou 1 rótulo). Em seguida, a

acurácia sofre uma queda gradual porque à medida que surgem novas atividades, o NOHAR leva um tempo para aprender o novo padrão e, durante esse período, ele comete alguns erros. A queda gradual da acurácia é típica de um algoritmo *online* quando o modelo de classificação não é pré-treinado com uma base de dados inicial, como ocorre na fase *offline* do algoritmo AnyNovel. No entanto, essa queda é temporária até o modelo de classificação conhecer bem o novo padrão após os refinamentos realizados na etapa de aprendizagem. Os experimentos descritos na Seção 7.4.3.2 mostram esse fenômeno nas bases de dados WISDM e UCI-HAR.

Em relação ao número de solicitações ao módulo de aprendizagem ativa. No total, foram 95 requisições na base de dados Shoaib, 32 para a atividade andar (barra pontilhada com valor 75) e 63 requisições para as demais atividades (barra pontilhada com valor 60). A próxima seção mostra a comparação dos números de requisições entre o NOHAR e o AnyNovel.

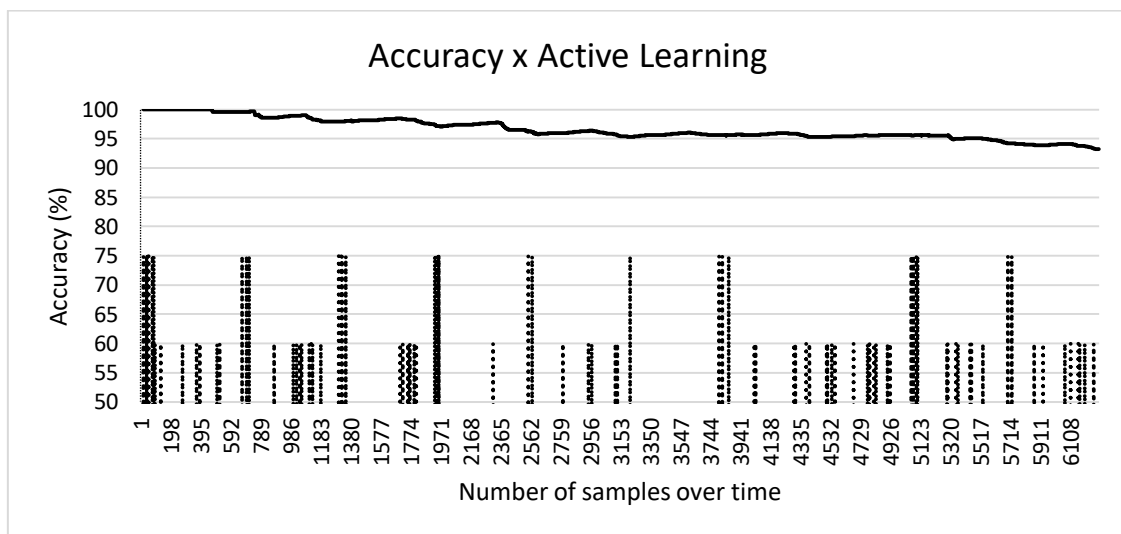


Figura 7.2 Taxa de acurácia do NOHAR ao longo do tempo e dos momentos em que o usuário é solicitado para fornecer o rótulo da atividade. A barra pontilhada que atinge o valor 75 indica uma requisição para a atividade andar e a barra pontilhada que atinge o valor 60 indica as requisições para as demais atividades.

#### 7.4.3.2 NOHAR x AnyNovel

Essa seção apresenta uma análise comparativa entre os algoritmos NOHAR e AnyNovel para as bases de dados SHOAIB, WISDM e UCI-HAR. Os resultados são apresentados em três partes. A primeira parte consiste em uma análise comparativa dos modelos de classificação em termos de assertividade. A segunda parte consiste em analisar o número de requisições que o algoritmo faz ao usuário por meio da aprendizagem ativa. Por fim, a terceira parte consiste em uma análise comparativa do tempo que os algoritmos levam para processar as bases de dados.

#### Acurácia dos Modelos de Classificação

Inicialmente, esta seção apresenta os resultados do algoritmo AnyNovel somente para a base de dados Shoaib. O objetivo é mostrar a acurácia do modelo de classificação de forma isolada como foi mostrado na Seção 7.4.3.1 para o algoritmo NOHAR. Em seguida, esta seção apresenta uma análise comparativa entre o NOHAR e o AnyNovel.

A Tabela 7.10 mostra um resumo das acurácias do modelo de classificação gerado pelo AnyNovel separadas por usuários e atividades. Como requisito do AnyNovel, os dados do usuário S1 foram utilizados para treino na fase *offline* e os restantes dos dados foram utilizados na fase *online*. Como pode ser observado, as atividades *Standing*, *Downstairs* e *Upstairs* são as atividades que o algoritmo AnyNovel possui maior dificuldade em aprender. A matriz de

confusão apresentada na Tabela 7.11 mostra que essas atividades confundem bastante o modelo de classificação. Além disso, o modelo confunde a atividade *Upstairs* com as atividades *Jogging* e *Walking*. Por outro lado, as atividades *Sitting*, *Biking* e *Walking* são as atividades em que o AnyNovel possui maior facilidade para aprender, a atividade *Biking*, por exemplo, obteve uma taxa de acurácia de 100% para o usuário S6. Em média, a taxa de acurácia para todos os usuários foi de 45,68%.

Tabela 7.10 Taxas de acurácia do algoritmo AnyNovel para cada usuário da base de dados SHOAB.

User	Accuracy	Walk	Standing	Jogging	Sitting	Biking	Upstairs	Downstairs
S1	-	-	-	-	-	-	-	-
S2	33.68	86.93	0.00	0.00	74.29	95.49	32.33	0.00
S3	35.14	92.87	0.00	82.00	71.90	48.98	8.01	0.00
S4	40.46	65.12	0.00	10.78	90.78	77.29	50.89	2.41
S5	29.01	77.77	0.00	0.00	83.33	77.62	12.94	0.00
S6	68.97	68.98	0.00	81.66	0.00	100.00	52.33	0.00
S7	73.62	73.62	0.00	0.00	78.79	80.80	22.06	0.00
S9	32.06	86.72	0.00	86.38	0.00	65.49	40.51	0.00
S10	52.56	52.57	46.33	33.23	61.86	7.99	7.41	0.00

A baixa taxa de acurácia do algoritmo AnyNovel está relacionada com as características fundamentais dos *clusters* gerados pelo AnyNovel. Os *clusters* possuem um formato de circunferência devido a medida de *Raio* utilizada como características do modelo de classificação [26]. A desvantagem dessa estratégia é que as fronteiras dos *clusters* não são flexíveis, permitindo a ocorrência de interseções desnecessárias entre os *clusters* no espaço. Além disso, a matriz de confusão mostra que apenas 0.37% (2119 de 558975) das amostras são classificadas como desconhecidas, os restantes são classificadas como alguma atividade existente ou uma atividade de novidade.

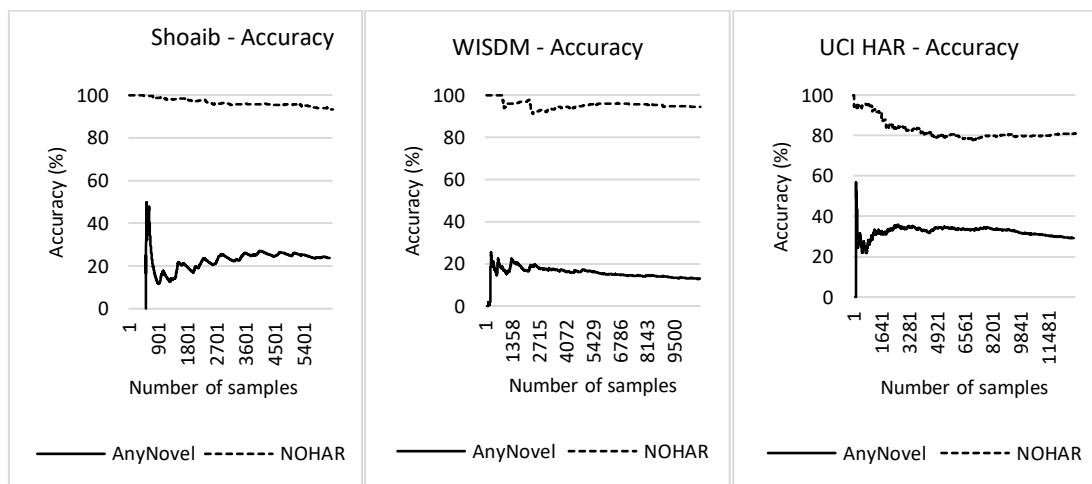
Tabela 7.11 Matriz de confusão do algoritmo AnyNovel.

	Walk	Standing	Jogging	Sitting	Biking	Upstairs	Downstairs	Unknown
Walk	7479	7479	34061	2092	7936	15506	637	0
Standing	13699	15621	21427	0	9103	10262	10748	0
Jogging	4351	9584	43483	869	681	20732	75	1183
Sitting	987	8975	35311	34623	254	0	0	936
Biking	48	9502	23623	3775	33850	7433	2768	0
Upstairs	0	19217	36610	0	3511	18903	2828	0
Downstairs	0	15781	45158	0	3692	14659	1642	0

A principal diferença entre o NOHAR e o AnyNovel é que o NOHAR gera modelos de classificação descritivos onde os padrões são capturados com base na estrutura dos histogramas. Isso significa que os padrões conseguem ser capturados mesmo quando existe intersecção entre o conjunto de dados originais. Isso melhora a capacidade do modelo de classificação de distinguir as classes. Por exemplo, as atividades estáticas *Sitting* e *Standing* são atividades similares onde os dados originais possuem intersecção entre si no espaço. Como o AnyNovel é baseado em *clusters* dos dados originais, o modelo de classificação se confunde bastante com essas atividades (Tabela 7.11). Por outro lado, os histogramas gerados pelo NOHAR conseguem diferenciar 100% ambas as atividades (Tabela 7.7). Outra diferença entre ambos os algoritmos é

que o AnyNovel avalia amostra por amostra, enquanto que o NOHAR avalia um conjunto de amostras representadas pelos histogramas.

Para finalizar a discussão entre AnyNovel e NOHAR, a Figura 7.3 mostra uma visão geral da análise comparativa das taxas de acurácia dos modelos de classificação obtidas ao longo tempo por ambos algoritmos. Como pode ser observado, o NOHAR é superior ao AnyNovel, em média, 70.12%, 72.99% e 68.30%, respectivamente para as bases de dados SHOAIB, WISDM e UCI-HAR. Dessa forma, é possível concluir que o AnyNovel não é adequado para atuar no contexto de HAR onde existe diversidades de atividades similares. Por outro lado, o NOHAR consegue superar os problemas do AnyNovel devido a característica do seu modelo de classificação aprender padrões baseados na estrutura dos histogramas, independente se os dados originais possuem intersecções ou não.



**Figura 7.3** Visão geral das acurácias do NOHAR e AnyNovel ao longo do tempo para as bases de dados SHOAIB, WISDM e UCI-HAR.

### Análise da Aprendizagem Ativa

Esta seção apresenta uma visão geral dos momentos em que os algoritmos NOHAR e AnyNovel solicitam informação do usuário para fornecer os rótulos das atividades por meio da aprendizagem ativa. Para esses experimentos, as iterações com os usuários foram simuladas com base na consulta aos rótulos disponíveis nas bases de dados, ou seja, quando um conceito novo é detectado, o algoritmo consulta o rótulo dos dados referente aquele momento.

A Figura 7.4 mostra o resumo dos resultados das taxas de acurácia ao longo do tempo juntamente com os momentos em que os modelos de classificação solicitaram o rótulo das atividades (barras pontilhadas). Para facilitar a visualização do número de requisições por atividade, os gráficos do NOHAR destacam o número de requisições para todas as atividades com destaque para a atividade andar representada pelas barras pontilhadas que atingem o valor 75. Por outro lado, o algoritmo AnyNovel não fornece informações para quais atividades as requisições foram direcionadas.

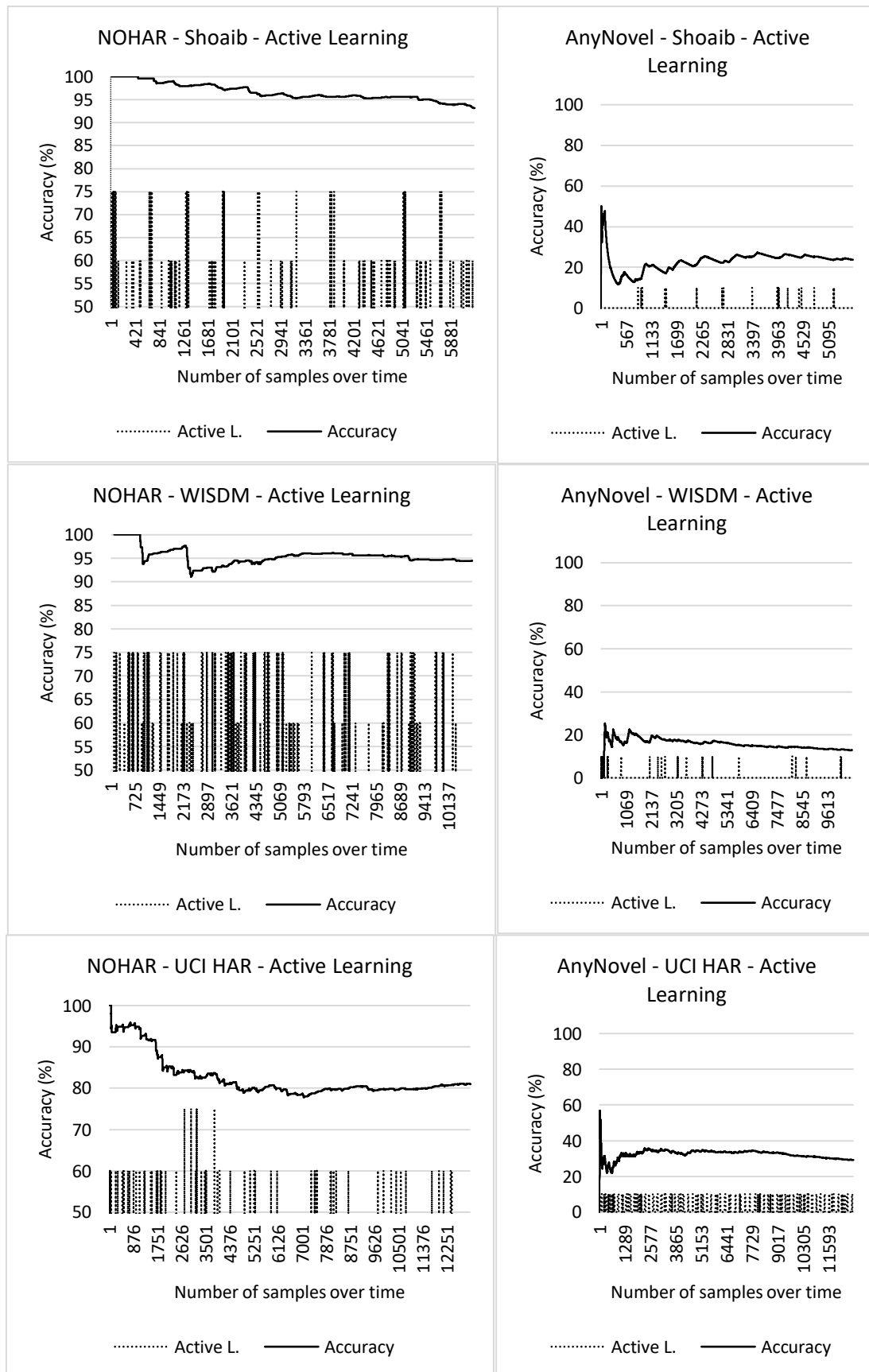


Figura 7.4 Acurácias (linhas superiores dos gráficos) e momentos em que o modelo de classificação solicitou o rótulo das atividades para o usuário (barras pontilhadas) dos algoritmos NOHAR e AnyNovel nas bases de dados SHOAI, WISDM e UCI-HAR.

Como pode ser observado na Figura 7.4, o NOHAR obteve o total de 95 (32 – andar) solicitações de rótulo contra 30 solicitações do AnyNovel para a base de dados Shoaib. De modo similar, o NOHAR obteve 161 (115 – andar) solicitações contra 33 do AnyNovel para a base de dados WISDM. Esses resultados mostram que o algoritmo AnyNovel é mais estável que o NOHAR porque os *clusters* que representam as atividades possuem bastante intersecções entre si. Por outro lado, o NOHAR é mais tolerante a *outliers*, visto que a probabilidade de um histograma contaminado com *outlier* se transformar em uma novidade é baixa, como pode ser visto nos resultados da base de dados UCI-HAR. O AnyNovel é sensível a *outlier* porque ele solicita o rótulo sempre que um dado desconhecido aparece.

### Tempo de Processamento

Esta seção apresenta os resultados referentes ao tempo de processamento dos algoritmos NOHAR e AnyNovel executados nas bases de dados SHOAIB, WISDM e UCI-HAR. Os experimentos foram realizados utilizando os dados de todos os usuários. A unidade de tempo utilizada é milissegundos (ms) e os resultados são baseados na média de tempo dos algoritmos executados 3 vezes. A Figura 7.5 mostra o resumo dos resultados de tempo de processamento para as respectivas bases de dados.

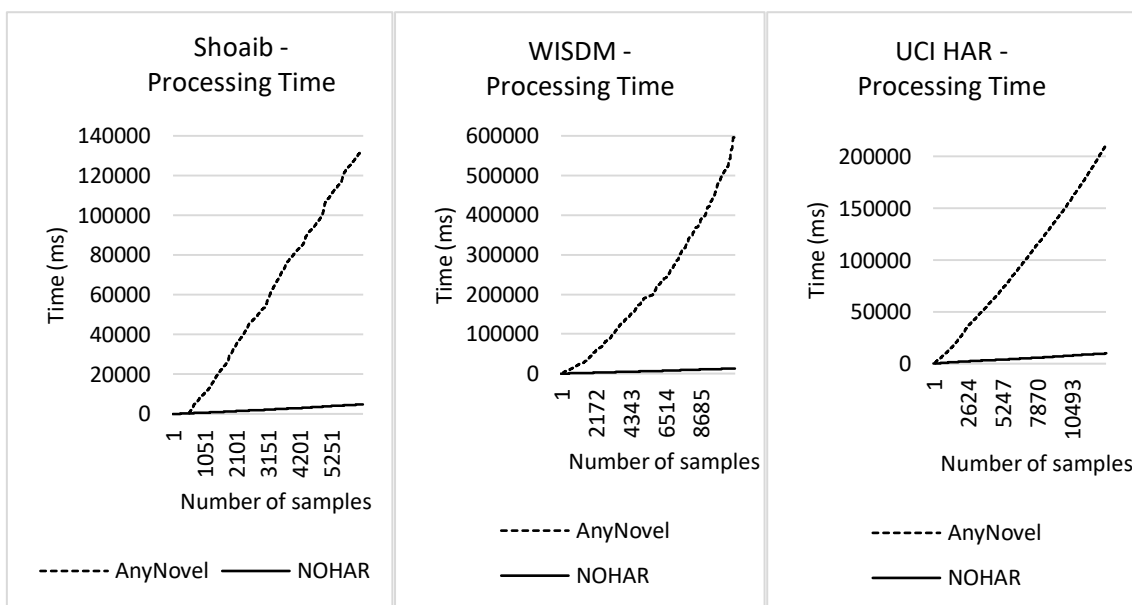


Figura 7.5 Visão geral do tempo de processamento dos algoritmos NOHAR e AnyNovel nas bases de dados SHOAIB, WISDM e UCI-HAR.

Como pode ser observado na Figura 7.5, o NOHAR levou 4901 ms, 12995 ms e 9912 ms e o AnyNovel levou 131782 ms, 598092 ms e 209790 ms para executar nas bases de dados SHOAIB, WISDM e UCI-HAR, respectivamente. Em resumo, isso mostra que o NOHAR é mais rápido que o algoritmo AnyNovel, em média, 96.28%, 97.82% e 95.27%, respectivamente.

Embora as complexidades de ambos algoritmos sejam lineares no tamanho da entrada, o AnyNovel processa individualmente cada amostra de um *chunk*, enquanto que o NOHAR processa uma compactação dessas amostras na forma de distribuição de frequências das palavras de um histograma, além do custo da compactação do *chunk*. Em outras palavras, o AnyNovel processa como entrada de dados o número de instâncias  $n$  de um *chunk*. Em seguida, essas instâncias são comparadas com o número de *clusters*  $k$ , onde em cada cluster ocorre o número de iterações  $i$ . Dessa forma, a complexidade do AnyNovel no tamanho da entrada é  $O(nki)$ , desconsiderando a complexidade dos demais componentes que também são lineares

[26]. Do mesmo modo, a complexidade do NOHAR no tamanho da entrada também é  $O(nki)$ . Nesse caso,  $n$  é o número de palavras (e frequências) de um histograma,  $k$  é o número de histogramas e  $i$  é o número de iterações sobre os histogramas. Além disso, o custo da discretização de um *chunk* é  $O(n)$ , visto que o algoritmo utilizado é o SAX [38]. Dessa forma, o custo total do NOHAR é  $O(nki) + O(n)$ .

#### 7.4.3.3 NOHAR x Hoeffding Tree

Esta seção apresenta uma análise comparativa entre os algoritmos NOHAR, *Hoeffding Tree* (HT) e *Hoeffding Adaptive Tree* (HAT) para as bases de dados SHOAIB, WISDM e UCI-HAR. Os resultados são apresentados em duas partes. A primeira parte consiste em uma análise comparativa dos modelos de classificação em termos de assertividade e a segunda parte consiste em uma análise comparativa do tempo em que os algoritmos levam para processar as bases de dados.

#### Acurácia dos Modelos de Classificação

Esta seção apresenta os resultados dos algoritmos HT e HAT para todos os usuários da base de dados Shoaib. O objetivo é mostrar um panorama geral do desempenho desses algoritmos no contexto do reconhecimento das atividades de forma isolada. Em seguida, esta seção apresenta uma análise comparativa com os resultados do algoritmo NOHAR. A Tabela 7.12 e a Tabela 7.13 mostram um resumo dos resultados das árvores de decisões representados pelas taxas de acurácias separadas por usuários e atividades.

**Tabela 7.12** Taxas de acurácia do algoritmo Hoeffding Tree o para cada usuário da base de dados SHOAIB.

User	Accuracy	Walking	Standing	Jogging	Sitting	Biking	Upstairs	Downstairs
S1	87.90	99.71	99.74	86.65	99.92	96.45	65.83	43.75
S2	92.87	99.81	99.85	97.40	99.95	97.27	78.39	66.42
S3	91.18	99.77	99.82	92.49	99.96	96.90	75.88	58.49
S4	88.48	99.82	99.85	85.34	99.97	97.71	67.03	50.47
S5	87.63	99.50	99.54	86.62	99.70	92.47	68.28	44.40
S6	90.48	99.81	99.68	95.42	99.96	96.98	71.77	49.20
S7	89.21	99.81	99.50	88.93	99.78	96.01	68.29	56.90
S8	89.28	99.81	99.16	88.06	99.94	96.73	69.54	55.47
S9	90.76	99.77	99.74	94.01	98.07	93.52	75.21	63.26
S10	87.49	99.76	98.40	89.98	98.74	86.66	70.09	49.46

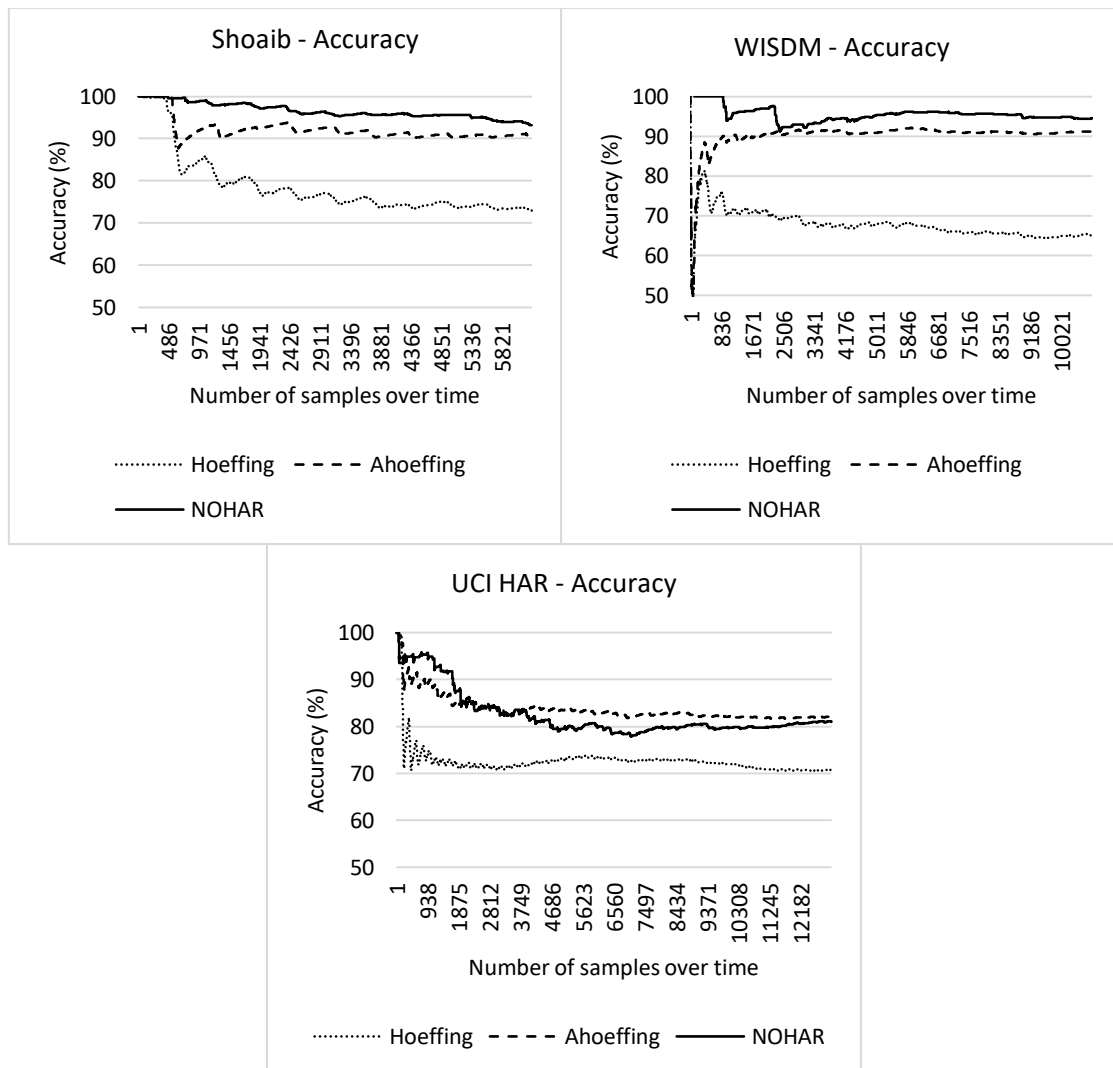
Como pode ser observado, as atividades *Upstairs* e *Downstairs* são as atividades que os algoritmos HT e HAT possuem maior dificuldade em aprender, assim como ocorreu com os resultados dos algoritmos NOHAR e AnyNovel descritos nas seções anteriores. Embora as acurácias da atividade *Downstairs* sejam maiores em HT e HAT, a média está em torno de 50%, isso significa que o algoritmo seleciona o rótulo de forma aleatória, sendo 50% de chance para acertar e errar rótulo. Isso mostra que o modelo de classificação não aprendeu os padrões dessa atividade.

A diferença entre os algoritmos HT e HAT, em termos de acurácia, é que o HT tem mais dificuldade de classificar a atividade *Jogging*, onde a acurácia média é de 90.49%, enquanto que o HAT possui uma média de 99.73%. Isso ocorre porque o algoritmo HT confunde a atividade *Jogging* com a atividade *Biking*. O desempenho dos restantes das atividades é similar ao obtido pelo algoritmo NOHAR, a diferença é que o NOHAR não

consegue aprender as atividades de alguns usuários devido a quantidade de dados não ser suficiente. Em média, as taxas de acurácia para os algoritmos HT e HAT foram de 73,07% e 90,97%, respectivamente.

**Tabela 7.13 Taxas de acurácia do algoritmo Hoeffding Adaptive Tree para cada usuário da base de dados SHOAI B.**

User	Accuracy	Walking	Standing	Jogging	Sitting	Biking	Upstairs	Downstairs
S1	91.14	99.71	99.75	99.82	99.92	99.77	72.03	37.28
S2	92.72	99.81	99.86	99.89	99.91	99.75	76.32	55.67
S3	91.38	99.80	99.81	99.87	99.91	99.81	72.58	39.97
S4	91.96	99.84	99.89	99.87	99.96	99.81	74.10	46.99
S5	91.16	99.59	99.58	99.65	99.72	99.61	72.29	39.51
S6	91.33	99.86	99.68	99.14	99.97	99.52	72.45	43.10
S7	91.64	99.85	99.82	99.76	99.81	99.69	73.34	44.02
S8	92.34	99.82	99.81	99.84	99.96	99.72	75.25	51.88
S9	92.76	99.84	99.83	99.77	99.10	98.77	77.14	59.30
S10	92.15	99.81	99.66	99.68	99.83	99.30	74.87	51.92



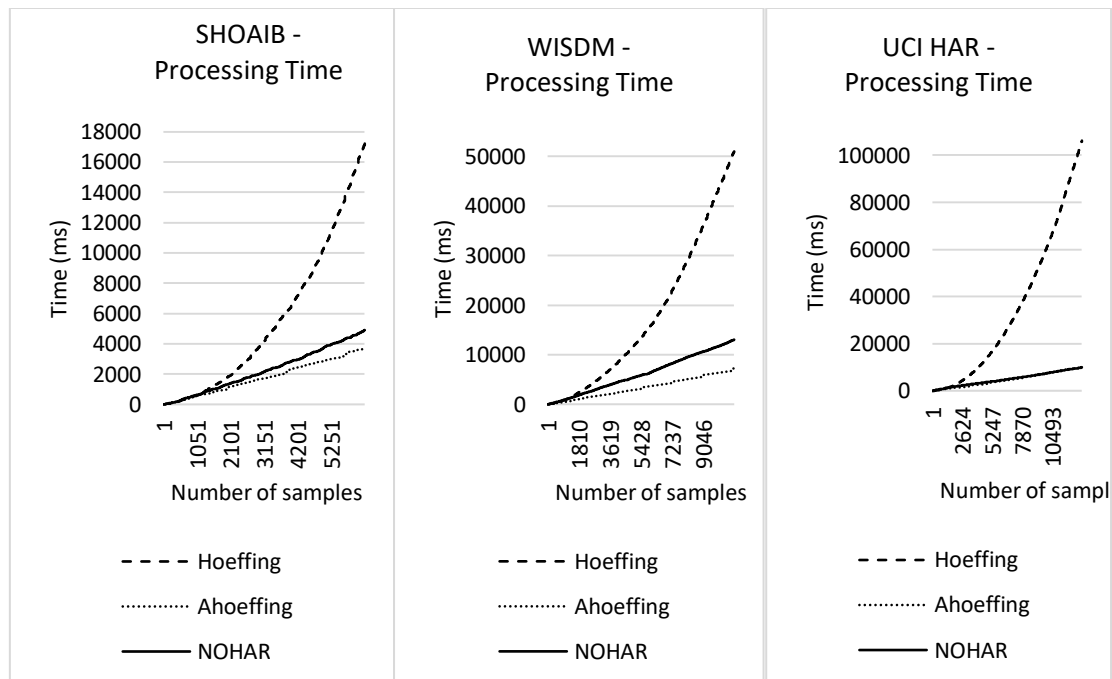
**Figura 7.6 Análise comparativas das acurácias ao longo do tempo dos algoritmos NOHAR, HT e HAT para as bases de dados SHOAI B, WISDM e UNI-HAR.**



Para finalizar a discussão entre NOHAR, HT e HAT, a Figura 7.6 mostra uma análise comparativa das taxas de acurácia dos modelos de classificação obtidas ao longo do tempo por esses algoritmos. Como pode ser observado, o algoritmo NOHAR conseguiu superar os algoritmos HT e HAT em todas as bases de dados, exceto na base de dados UCI-HAR onde o NOHAR apresentou taxas de acurácia inferior ao HAT. Nesse sentido, os resultados do HAT mostram a sua capacidade de lidar com ruídos melhor que o algoritmo NOHAR. O HAT é um algoritmo da categoria de evolução de dados que utiliza o algoritmo de detecção de mudanças ADWIN para decidir os momentos em que o modelo de classificação deve ser alterado ou não com base no surgimento de novos conceitos. No geral, a diferença dos resultados do HAT para o NOHAR é, em média, 3,49% para todas as bases de dados.

### Tempo de Processamento

Esta seção apresenta os resultados referentes ao tempo de processamento dos algoritmos NOHAR, HT e HAT executados nas bases de dados SHOAIB, WISDM e UCI-HAR. Os experimentos foram realizados utilizando os dados de todos os usuários. A unidade de tempo utilizada é milissegundos (ms) e os resultados são baseados na média de tempo dos algoritmos executados 3 vezes. A Figura 7.7 mostra o resumo dos resultados de tempo de processamento para as respectivas bases de dados.



**Figura 7.7** Visão geral do tempo de processamento dos algoritmos NOHAR, Hoeffding Tree e Hoeffding Adaptive Tree nas bases de dados SHOAIB, WISDM e UCI-HAR.

Como pode ser observado na Figura 7.7, o HT foi, em média, 6 vezes mais lento que o algoritmo NOHAR, sendo que a função de tempo forma uma curva não linear que indica que quanto maior a quantidade de dados, maior será o tempo de processamento na ordem da complexidade  $O(n \log n)$  no pior caso. Especificamente, a complexidade de HT é  $O(TAVC \log W)$ , onde  $T$  é o número de nós da árvore,  $A$  o número de atributos,  $V$  número máximo dos valores para um atributo,  $C$  número de classes e  $W$  tamanho da janela de tempo [36]. Como o algoritmo NOHAR possui complexidade linear  $O(nki) + O(n)$ , o tempo de processamento cresce na mesma proporção da entrada de dados, conforme mostra o resultado da função linear nos gráficos.

Por outro lado, o HAT foi mais rápido que o NOHAR nas bases de dados SHOAI B e WISDM. No entanto, a complexidade do HAT é a mesma complexidade do HT, a diferença é que o algoritmo ADWIN, com complexidade  $O(\log W)$ , reduz a quantidade de vezes em que a árvore de decisão é atualizada, sendo que ocorre atualização somente quando o ADWIN acusa mudança de conceito. Mesmo assim, pela complexidade do HAT ser  $O(n \log n)$ , em algum momento ao longo da análise de dados, o tempo de processamento do HAT ultrapassará o tempo de processamento do NOHAR. Uma percepção sobre isso pode ser visto nos resultados de tempo da base de dados UCI-HAR, onde o tempo de processamento do NOHAR é igual ao do HAT. O problema é que as bases de dados utilizadas nesses experimentos não possuem dados suficientes para mostrar esse comportamento.

#### 7.4.3.4 Análise do Consumo de Memória

Esta seção apresenta os resultados referentes ao consumo de memória antes e depois da discretização dos dados pelo algoritmo NOHAR nas bases de dados UCI-HAR, WISDM e SHOAI B. Os resultados estão descritos na Tabela 7.14 em forma da unidade de medida *Bytes*. A estratégia utilizada para extrair esses resultados foi a seguinte: 1) calcular a quantidade de *Bytes* que os dados brutos ocupam na memória; 2) aplicar o processo de discretização; e 3) calcular a quantidade de *Bytes* que os dados discretos ocupam na memória.

**Tabela 7.14** Consumo de memória em Bytes dos dados antes e depois do processo de extração de características do algoritmo NOHAR.

Dataset	Raw data	8 Letters	4 Letters
UCI-HAR	43,409,405	1,779,486	7,339
WISDM	22,035,537	866,972	5,995
SHOAIB	16,175,747	1,377,616	7,906

O NOHAR utiliza o algoritmo SAX para discretizar as séries temporais. O parâmetro referente ao tamanho das palavras influencia bastante no consumo da memória física dos dispositivos. Por exemplo, a Tabela 7.14 mostra os resultados ao SAX configurado com as palavras de tamanho 8 e tamanho 4. O parâmetro de tamanho 8 reduziu o espaço consumido da memória, em média, 94.48% para todas as bases de dados. Enquanto que o parâmetro de tamanho 4 reduziu o espaço consumido da memória, em média, 99.97% para todas as bases de dados. Na prática, isso significa uma redução de 43,409,405 *Bytes* referentes aos dados originais para 7,339 *Bytes* considerando apenas a base de dados UCI-HAR. No caso do NOHAR, o SAX está configurado com o tamanho da palavra igual a 4.

Isso mostra o quão capazes são os algoritmos de representação simbólica em compactar os dados e manter a eficiência dos modelos de classificação, além de permitir a manipulação de uma quantidade massiva de dados.

## 7.5 Considerações Finais

Este capítulo descreveu os resultados dos modelos de reconhecimento de atividades humanas em três cenários: 1) algoritmos *offline* rasos e profundos; 2) algoritmos de classificação simbólicos; e 3) algoritmos *online* incluindo o NOHAR. Os experimentos relacionados aos algoritmos *offline* rasos e profundos podem ser vistos no Apêndice C. Os resultados descritos neste capítulo geraram 1 artigo e 1 periódico, com previsões para publicações de mais dois

periódicos. Uma discussão detalhada sobre as principais conclusões deste capítulo está descrita no Capítulo 8.

# CAPÍTULO 8

## CONCLUSÕES E TRABALHOS FUTUROS

Esta tese de doutorado apresenta uma nova abordagem para reconhecimento de atividade humanas *online* e adaptativa baseada em smartphones instrumentados com sensores inerciais. A abordagem proposta é composta pelo *framework* DISTAR e pelo algoritmo simbólico *online* NOHAR, ambos atuam no contexto da análise de fluxo contínuo de dados discretos. A discretização dos dados coletados pelos sensores é baseada em algoritmos de representação simbólica. A principal vantagem desses algoritmos é a manipulação de uma quantidade massiva de dados com menor custo de processamento. Essa vantagem é devido a capacidade dos algoritmos simbólicos em reduzir a dimensionalidade e a numerosidade dos dados por meio da compactação de séries temporais. Nesse sentido, o NOHAR alcançou seu principal objetivo que é desenvolver uma solução adaptativa, escalável e de baixo custo capaz de ser implementada em smartphones. As principais conclusões extraídas dos experimentos são:

1. O NOHAR é capaz de aprender bem as atividades físicas humanas com uma taxa de acurácia exata ou próxima a 100%, exceto atividades que possuem a estrutura das séries temporais similares umas às outras onde o NOHAR tem dificuldade em aprender, como as atividades *Upstairs* e *Downstairs*.
2. O classificador NOHAR supera os algoritmos AnyNovel, *Hoeffing Tree* e *Hoeffing Adaptive Tree* em termos de acurácia. Além de ser, em média, 13 vezes mais rápido que os algoritmos avaliados.
3. O NOHAR reduz o consumo de memória dos dispositivos, em média, 99,97% para as bases de dados avaliadas considerando um alfabeto de 4 letras.
4. O NOHAR é um algoritmo de aprendizagem de máquina simbólico *online* inédito que está inserido na área de fluxo contínuo de dados.

Além disso, essa tese sugere o emprego de novos algoritmos *offline* baseados na representação simbólica dos dados para serem utilizados nas soluções de HAR. A vantagem desses algoritmos está também relacionada com a redução do custo da metodologia tradicional de HAR por meio da redução das etapas de processamento de dados, como a etapa de redução de ruído de dados e a etapa de redução de dimensionalidade dos dados. Os novos algoritmos de classificação de séries temporais sugeridos para serem utilizados na área de HAR são SAX-VSM, BOSS-VS e WEASEL. Todos eles são baseados nas técnicas de discretização SAX e SFA. Além disso, esta tese propõe duas soluções para tratamento de séries temporais multidimensionais devido à natureza dos dados dos sensores inerciais. A primeira solução consiste em uma matriz de segmentos sincronizados que permite o empilhamento de histogramas e a segunda solução consiste no emprego de técnicas de fusão de coordenadas como magnitude e PCA. As principais conclusões extraídas dos experimentos são:

- 1) O algoritmo BOSS-VS obteve as melhores taxas de acurácia usando as estratégias de fusão de coordenadas proposta neste trabalho.
- 2) O algoritmo Weasel não obteve um bom desempenho com as estratégias de fusão de coordenadas baseadas na magnitude e no PCA.

- 3) A estratégia de fusão de coordenadas baseada no empilhamento de histogramas gera os melhores classificadores.
- 4) O tempo de processamento da etapa de extração de características é mais rápido para as características do domínio discreto. Em média, essas características são 84,81% mais rápidas que as características dos domínios de tempo e frequência. Nesse caso, o SFA é o método com menor tempo de processamento, levando uma média de 418 milissegundos para uma quantidade de dados em torno de 1 milhão de amostras (base de dados WIDSM). A velocidade de processamento do NOHAR é devido a um processo de otimização baseado na transformação dos dados para *bits*.
- 5) O SAX-VSM é o algoritmo com menor tempo de execução nas etapas de extração de características e treinamento dos modelos de classificação, superando todos *baselines*. Por outro lado, os algoritmos BOSS-VS e Weasel são os algoritmos mais lentos na etapa de treinamento dos modelos de classificação.
- 6) As características do domínio discreto reduzem o consumo de memória em média 94,48% em comparação com as características do domínio de frequência, com 92,66%, e as características do domínio de tempo, com 73,90%. Embora a diferença entre as características dos domínios de frequência e discreto sejam de apenas 1,92%, as características do domínio da frequência perdem em termos de acurácia do modelo de classificação e tempo de processamento.

Além disso, este trabalho fornece uma visão geral do estado da arte das soluções de HAR no contexto dos smartphones com sensores inerciais. O estado da arte contém discussões baseadas na apresentação dos conceitos e metodologias utilizadas para reconhecer as atividades humanas. Esta tese apresentou um histórico completo, focado nos smartphones, que mostra a evolução da área nas últimas duas décadas. Aplicações, desafios e oportunidades para futuras pesquisas na área de HAR baseada em smartphones foram apresentadas. Em seguida, foi apresentada uma discussão das diferenças entre as abordagens baseadas na aprendizagem com algoritmos profundos e rasos. Os resultados mostram que os algoritmos de aprendizagem profundos são capazes de gerar modelos de classificação com alta acurácia. Porém, o uso desses algoritmos no contexto dos smartphones ainda é um desafio devido esses dispositivos possuírem recursos computacionais limitados como memória, processamento e bateria. As comparações entre as soluções do estado da arte mostram que os modelos de classificação gerados com as características do domínio do tempo são os melhores, com destaque para os algoritmos da família das árvores de decisões, em especial, o algoritmo *Random Forest*. Por fim, os resultados mostram que a fusão de dados entre sensores melhora a qualidade dos modelos de classificação.

As limitações do NOHAR podem ser resumidas em dois pontos principais. O primeiro ponto trata da dificuldade do NOHAR em aprender padrões de atividades quando não existem dados suficientes. Isso implica que o NOHAR leva mais tempo para aprender um padrão porque, esse padrão, deve ser confirmando ao longo do tempo, ou seja, ele deve reaparecer considerando um limite mínimo de observações definidas como parâmetros. O segundo ponto trata da definição de uma grande quantidade de parâmetros incluindo tamanhos de janela de tempo, palavras, alfabeto e BOP. Se o NOHAR for aplicado em uma área diferente de HAR, provavelmente os parâmetros não serão os mesmos e, portanto, devem ser definidos com bases em estudos e particularidades que fazem sentido para o problema em questão.

## 8.1 Trabalhos futuros

Esta seção apresenta uma lista dos possíveis trabalhos futuros que podem ser explorados nas próximas pesquisas. Nesta ocasião, é possível vislumbrar alguns trabalhos futuros no contexto dos algoritmos *offline*:

1. Desenvolver um algoritmo de representação simbólica que utiliza os conhecimentos de processamento natural de linguagem com o propósito de substituir a representação de dados de histogramas por gramáticas utilizando, por exemplo, os algoritmos *Repair* [220] e *Sequitur* [221].
2. Explorar soluções de baixo custo baseadas em algoritmos de aprendizagem profunda como a rede *Mobile Net* [222].

No contexto dos algoritmos *online*, as sugestões são:

3. Utilizar transferência de conhecimento entre modelo de classificação com o propósito de facilitar a experiência de usuários com o mesmo perfil.
4. Explorar o NOHAR em outros domínios de aplicações com o propósito de disseminar os algoritmos *online* simbólicos.

Outras sugestões que poderiam ser exploradas são:

5. Estudos na etapa de segmentação do sinal analisando tamanho de janelas variáveis.
6. Soluções de HAR exploram a fusão de dados de outros sensores como os de localização e os de rede.
7. Soluções de HAR na área de segurança onde as atividades podem ser utilizadas na autenticação dos usuários.

Além disso, o NOHAR ainda precisa ser implementado nos smartphones para que ele seja avaliado em um ambiente real.

## 8.2 Publicações

Esta proposta parcial de tese gerou as seguintes publicações:

### Periódicos publicados (Qualis A1 - CAPES):

1. SOUSA LIMA, WESLLEN; DE SOUZA BRAGANÇA, HENDRIO; MONTERO QUISPE, KEVIN; PEREIRA SOUTO, EDUARDO. Human Activity Recognition Based on Symbolic Representation Algorithms for Inertial Sensors. *SENSORS*, v. 18, p. 4045-4045, 2018.
2. MONTERO QUISPE, KEVIN; SOUSA LIMA, WESLLEN; MACÊDO BATISTA, DANIEL; SOUTO, EDUARDO. MBOSS: A Symbolic Representation of Human Activity Recognition Using Mobile Sensors. *SENSORS*, v. 18, p. 4354, 2018.

### Capítulos de livro publicados:

3. Andrade, Thiago; Ribeiro, Rita P; Sousa, Wesllen; Carvalho, André; and Gama, João. Anomaly Detection in Sequential Data: Principles and Case Studies. In: *Wiley Encyclopedia of Electrical and Electronics Engineering*. 2018.

### Artigo publicados:

4. SOUSA, WESLLEN; SOUTO, EDUARDO; RODRIGRES, JONATAS; SADARC, PEDRO; JALALI, ROOZBEH; EL-KHATIB, KHALIL. A Comparative Analysis of the Impact of Features on Human Activity Recognition with Smartphone Sensors. In: 23rd Brazillian Symposium on Multimedia and the Web, 2017, Gramado-RG. New York: ACM Press, 2017. p. 397-404.
5. QUISPE, KEVIN G. MONTERO; LIMA, WESLLEN SOUSA; SOUTO, EDUARDO J. PEREIRA. Human Activity Recognition on Smartphones using Symbolic Data Representation. In: the 24th Brazilian Symposium, 2018, Salvador. Proceedings of the 24th Brazilian Symposium on Multimedia and the Web - WebMedia '18. New York: ACM Press, 2018. p. 93.

**Outros artigos publicados durante o período do doutorado:**

6. JUNIOR, NAMEDIN; SOUTO, EDUARDO; SOUSA, WESLLEN; NASCIMENTO, ROGÉRIO P.C.; ASCHOFF, RAFAEL. A Visual Approach for the Definition of Behavior in Environments for Internet of Things. In: 23rd Brazillian Symposium on Multimedia and the Web, 2017, Gramado. New York: ACM Press, 2017. p. 381.
7. SOUSA, WESLLEN; SOUTO, EDUARDO; ROCHA, THIAGO; PEREIRA, NAMEDIN; KOOL, PEETER; ROSENGREN, PETER. Prediction of electrical energy consumption for Internet of Things in disaggregated databases. In: 2016 IEEE Symposium on Computers and Communication (ISCC), 2016, Messina. 2016. p. 429.
8. LIMA, WESLLEN S.; SOUTO, EDUARDO; ROCHA, THIAGO; PAZZI, RICHARD W.; PRAMUDIANTO, FERRY. User activity recognition for energy saving in smart home environment. In: 2015 IEEE Symposium on Computers and Communication (ISCC), 2015, Larnaca. v. 20. p. 751-828.

## BIBLIOGRAFIA

- [1] J. W. Lockhart, T. Pulickal e G. M. Weiss, "Applications of mobile activity recognition," *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, 2012.
- [2] W. Z. Khan, Y. Xiang, M. Y. Aalsalem e Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Communications Surveys and Tutorials*, vol. 15, nº 1, pp. 402-427, 2013.
- [3] J. Dai, X. Bai, Z. Yang, Z. Shen e D. Xuan, "PerFallD: A pervasive fall detection system using mobile phones," *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 292-297, 2010.
- [4] J. Fontecha, F. J. Navarro, R. Hervás e J. Bravo, "Elderly frailty detection by using accelerometer-enabled smartphones and clinical information records," *Personal and Ubiquitous Computing*, vol. 17, nº 6, pp. 1073-1083, 2013.
- [5] D. Preuveneers e Y. Berbers, "Mobile phones assisting with health self-care: a diabetes case study," *10th international conference on Human computer interaction with mobile devices and services*, pp. 177-186, 2008.
- [6] E. M. Tapia, S. S. Intille e K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," *International Conference on Pervasive Computing*, pp. 158-175, 2004.
- [7] W. S. LIMA, E. SOUTO, T. ROCHA, R. W. PAZZI e F. PRAMUDIANTO, "User activity recognition for energy saving in smart home environment," *IEEE Symposium on Computers and Communication (ISCC)*, pp. 751-828, 2015.
- [8] D. Bonino e F. Corno, "DogOnt - Ontology Modeling for Intelligent Domotic Environments," *Energy and Buildings*, pp. 790-803, 2008.
- [9] O. A. Postolache, J. D. Pereira e P. S. Girão, "Smart sensors network for air quality monitoring applications," *IEEE Transactions on Instrumentation and Measurement*, pp. 3253-3262, 2009.
- [10] W. Niu, J. Long, D. Han e Y. F. Wang, "Human activity detection and recognition for video surveillance," *IEEE International Conference on Multimedia and Exp (Icme)*, Vols. %1 de %21-3, pp. 719-722, 2004 .
- [11] R. Mehran, A. Oyama e M. Shah, "Abnormal crowd behavior detection using social force model," *Computer Vision and Pattern Recognition*, pp. 935-94, 2009.
- [12] P. Viola e M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 511-518, 2001.
- [13] D. Choujaa e N. Dulay, "Activity Recognition from Mobile Phone Data: State of the Art, Prospects and Open Problems," *Imperial College London*, vol. 5, p. 32, 2009.
- [14] L. Liao, D. J. Patterson, D. Fox e H. Kautz, "Learning and inferring transportation



routines," *Artificial Intelligence*, vol. 171, n° 5-6, pp. 311-331, 2007.

- [15] S. Saeedi, "Context-Aware Personal Navigation Services Using Multi-level Sensor Fusion Algorithms," *PhD Thesis*, 2013.
- [16] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, Z. Yu e S. Member, "Sensor-Based Activity Recognition," *Systems, man, and cybernetics, Part C: Applications and reviews*, vol. 42, pp. 790-808, 2012.
- [17] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury e A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, pp. 140-150, 2010.
- [18] O. D. Incel, M. Kose e C. Ersoy, "A Review and Taxonomy of Activity Recognition on Mobile Phones," vol. 3, pp. 145-171, 2013.
- [19] M. M. J. B. G. a. F. S. Gaber, *Pocket data mining. Big Data on Small Devices.*, Springer, 2014.
- [20] D. J. Cook e N. C. Krishnan, *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data.*, John Wiley & Sons, 2015.
- [21] M. F. A. b. Abdullah, F. P. N. Ali, M. S. Sayeed, D.-J. Choi e K. S. Muthu, "Classification algorithms in human activity recognition using smartphones," *International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering*, vol. 6, n° 8, pp. 362-369, 2012.
- [22] O. D. Incel, M. Kose e C. Ersoy, "A Review and Taxonomy of Activity Recognition on Mobile Phones," *BioNanoScience*, vol. 3, pp. 145-171, 2013.
- [23] R. Khusainov, D. Azzi, I. E. Achumba e S. D. Bersch, "Real-time human ambulation, activity, and physiological monitoring: taxonomy of issues, techniques, applications, challenges and limitations.," vol. 13, n° 10, pp. 12852-12902, 2013.
- [24] J. Wang, Y. Chen, S. Hao, X. Peng e L. Hu, "Deep Learning for Sensor-based Activity Recognition: A Survey," *Computer Vision and Pattern Recognition*, 2017.
- [25] Z. S. Abdallah, M. M. Gaber, B. Srinivasan e S. Krishnaswamy, "Adaptive mobile activity recognition system with evolving data streams," *Neurocomputing*, vol. 150, pp. 304-317, 2015.
- [26] Z. S. Abdallah, M. M. Gaber, B. Srinivasan e S. Krishnaswamy, "AnyNovel: detection of novel concepts in evolving data streams," *Evolving Systems*, vol. 7, n° 2, pp. 73-93, 2016.
- [27] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten e P. J. Havinga, "Complex human activity recognition using smartphone and wrist-worn motion sensors," *Sensors (Switzerland)*, vol. 16, n° 4, pp. 1-24, 2016.
- [28] Y. Chen e C. Shen, "Performance Analysis of Smartphone-Sensor Behavior for Human Activity Recognition," *IEEE Access*, vol. 5, pp. 3095-3110, 2017.
- [29] J. Yang, "Toward Physical Activity Diary : Motion Recognition Using Simple Acceleration Features with Mobile Phones," *Proceedings of the 1st international workshop on*

*Interactive multimedia for consumer electronics*, pp. 1-9, 2009.

- [30] I. Anderson, J. Maitland, S. Sherwood, L. Barkhuus, M. Chalmers, M. Hall, B. Brown e H. Muller, "Shakra: Tracking and sharing daily activity levels with unaugmented mobile phones," *Mobile Networks and Applications*, vol. 12, nº 2-3, pp. 185-199, 2007.
- [31] M. Berchtold, M. Budde, D. Gordon, H. Schmidtke e M. Beigl, "ActiServ: Activity Recognition Service for mobile phones," *International Symposium on Wearable Computers*, pp. 1-8, 2010 .
- [32] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu e A. T. Campbell, "Darwin Phones : the Evolution of Sensing and Inference on Mobile Phones," *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. pp. 5-20, 2010.
- [33] Z. Zhao, Z. Chen, Y. Chen, S. Wang e H. Wang, "A Class Incremental Extreme Learning Machine for Activity Recognition," *Cognitive Computation*, vol. 6, nº 3, pp. 423-431, 2014.
- [34] C. Hu, Y. Chen, L. Hu e X. Peng, "A novel random forests based class incremental learning method for activity recognition," *Pattern Recognition*, vol. 78, pp. 277-290, 2018.
- [35] J. Gama, P. P. Rodrigues, E. Spinoso e A. Carvalho, "Knowledge Discovery from Data Streams," *Web Intelligence and Security - Advances in Data and Text Mining Techniques for Detecting and Preventing Terrorist Activities on the Web*, pp. 125-138, 2010.
- [36] A. Bifet, G. Holmes, R. Kirkby e B. Pfahringer, "Data Stream Mining: A Practical Approach," *COSI - Centre for Open Software Innovation*, pp. 1-179, 2011.
- [37] D. Figo, P. C. Diniz, D. R. Ferreira e J. M. P. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, nº 7, pp. 645-662, 2010.
- [38] J. Lin, E. Keogh, L. Wei e S. Lonardi, "Experiencing SAX: a Novel Symbolic Representation of Time Series," *Data Mining and Knowledge Discovery* , vol. 15 , nº 2, pp. 107 - 144, 2007 .
- [39] P. Schäfer e M. Höggqvist, "SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets," *Data Mining and Knowledge Discovery*, vol. 29, pp. 228-242, 2012.
- [40] P. Senin e S. Malinchik, "SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model," *IEEE 13th International Conference on Data Mining*, vol. 44, nº 0704, pp. 1175-1180, 2013 .
- [41] P. Schäfer, "Bag-Of-SFA-Symbols in Vector Space ( BOSS VS )," *ZIB Report*, pp. 15-30, 2015.
- [42] P. Schäfer, "The BOSS is concerned with time series classification," *Data Mining and Knowledge Discovery*, vol. 6, nº 29, pp. 1505-1530, 2015.

- [43] P. Schäfer e L. Ulf, "Fast and Accurate Time Series Classification with WEASEL," *arXiv preprint arXiv:1701.07681*, 2017.
- [44] P. Siirtola, H. Koskimäki, V. Huikari, P. Laurinen e J. Röning, "Improving the classification accuracy of streaming data using SAX similarity features," *Pattern Recognition Letters*, vol. 32, nº 13, pp. 1659-1668, 2011.
- [45] D. Figo, P. C. Diniz, D. R. Ferreira e J. M. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, nº 7, pp. 645-662, 2010.
- [46] J. Gama e C. Pinto, "Discretization from data streams: Applications to histograms and data mining," *ACM Symposium on Applied Computing*, p. 662–667, 2006.
- [47] G. H. a. P. Domingos, "Mining complex models from arbitrarily large databases in constant time," *International Conference on Knowledge Discovery and Data Mining*, p. 525–531, 2002.
- [48] J. Shieh e E. Keogh, "iSAX: Indexing and Mining Terabyte Sized Time Series," *14th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 623, 2008.
- [49] W. Sousa Lima, H. De Souza Bragança, K. Montero Quispe e E. Pereira Souto, "Recognition Based on Symbolic Representation Algorithms for Inertial Sensors," *Sensors*, vol. 18, p. 4045, 2018.
- [50] K. Montero Quispe, W. Sousa Lima, D. Macêdo Batista e Souto, "MBOSS: A Symbolic Representation of Human Activity Recognition Using Mobile Sensors," *Sensors*, p. 4354, 18.
- [51] K. G. M. Quispe, W. S. Lima e E. J. P. Souto, "Human Activity Recognition on Smartphones using Symbolic Data Representation," *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, pp. 93-100, 2018.
- [52] T. Andrade, J. Gama, R. P. Ribeiro, W. Sousa e A. d. Carvalho, *Anomaly Detection in Sequential Data: Principles and Case Studies*, Wiley Encyclopedia of Electrical and Electronics Engineering, 2019.
- [53] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas e D. J. Cook, "Simple and Complex Activity Recognition through Smart Phones," *2012 Eighth International Conference on Intelligent Environments*, pp. 214-221, 2012.
- [54] A. M. Khan, A. Tufail, A. M. Khattak, A. M. Khattak e T. H. Laine, "Activity recognition on smartphones via sensor-fusion and KDA-based SVMs," *International Journal of Distributed Sensor Networks*, 2014.
- [55] A. Reiss, "Personalized Mobile Physical Activity Monitoring for Everyday Life," 2013.
- [56] M. Li, P. Zhou, Y. Zheng, Z. Li e G. Shen, "IODetector: A Generic Service for Indoor/Outdoor Detection," *ACM Trans. Sensor Network*, vol. 11, nº 2, pp. 28:1--28:29, 2014.

- [57] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen e M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks*, vol. 6, nº 2, pp. 1-27, 2010.
- [58] M. Weiser, The computer for the 21st century, *Scientific american*, 1991, pp. 94-104.
- [59] C. Ramos, J. C. Augusto e D. Shapiro, "Ambient intelligence - the next step for artificial intelligence," *Intelligent Systems*, pp. 15-18, 2008.
- [60] W. S. Lima, E. Souto, T. Rocha e R. W. Pazzi, "User activity recognition for energy saving in smart home environment," *IEEE Symposium on Computers and Communication (ISCC)*, pp. 751 - 757, 2015.
- [61] C. Orwat, A. Graefe e T. Faulwasser, Towards pervasive computing in health care - A literature review, *BMC Medical Informatics and Decision Making*, 2008.
- [62] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King e R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," *Wearable Computers, 2007 11th IEEE International Symposium*, pp. 37-40, 2007.
- [63] S. Das, L. Green, B. Perez, M. Murphy e A. Perring, "Detecting user activities using the accelerometer on Android smartphones," *The Team for Research in Ubiquitous Secure Technology*, 2010.
- [64] G. Bieber, P. Koldrack, C. Sablowski, C. Peter e B. Urban, "Mobile physical activity recognition of stand-up and sit-down transitions for user behavior analysis," 2010.
- [65] T. Huynh, M. Fritz e B. Schiele, "Discovery of activity patterns using topic models," *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, p. 10, 2008.
- [66] V. Srinivasan, S. Moghaddam, A. Mukherji, K. K. Rachuri, C. Xu e E. M. Tapia, "MobileMiner: Mining Your Frequent Patterns on Your Phone," *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '14 Adjunct*, pp. 389-400, 2014.
- [67] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng e A. T. Campbell, "Sensing Meets Mobile Social Networks: The Design , Implementation and Evaluation of the CenceMe Application," *SenSys '08*, pp. 337-350, 2008.
- [68] J. Gong, J. Tang e A. Fong, "ACTPred: Activity prediction in mobile social networks," *Tsinghua Science and Technology*, vol. 29, nº 3, pp. 265-274, 2014.
- [69] X. Hu, T. H. Chu, H. C. Chan e V. C. Leung, "Vita: A Crowdsensing-Oriented Mobile Cyber-Physical System," *IEEE Transactions on Emerging Topics in Computing*, pp. 148-165, 2013.
- [70] F. Xia, L. T. Yang, L. Wang e A. & Vinel, "Internet of things," *International Journal of Communication Systems*, 2012.

- [71] H. Chourabi, T. Nam, S. Walker, J. Gil-Garcia, S. Mellouli, K. Nahon, T. Pardo e H. Scholl, "Understanding smart cities: An integrative framework," *System Science (HICSS), 2012 45th Hawaii International Conference*, pp. 2289-2297, 2012.
- [72] T. Sohn, A. Varshavsky, A. Lamarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold e E. D. Lara, "Mobility Detection Using Everyday GSM Traces," *UbiComp*, pp. 212-224, 2006.
- [73] T. Iso e K. Yamazaki, "Gait analyzer based on a cell phone with a single three-axis accelerometer," *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services - MobileHCI '06*, p. 141, 2006.
- [74] T. Saponas, J. Lester, J. Froehlich, J. Fogarty e J. Landay, "iLearn on the iPhone: Real-Time Human Activity Classification on Commodity Mobile Phones," *University of Washington CSE Tech Report UWCSE080402*, 2008.
- [75] H. Lu, W. Pan, N. Lane, T. Choudhury e A. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones," *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pp. 165-178, 2009.
- [76] J. Ryder, B. Longstaff, S. Reddy e D. Estrin, "Ambulation: A Tool for Monitoring Mobility Patterns over Time Using Mobile Phones," *2009 International Conference on Computational Science and Engineering*, vol. 4, pp. 927-931, 2009.
- [77] S. Purpura, V. Schwanda, K. Williams, W. Stubler e P. Sengers, "Fit4life: The design of a persuasive technology promoting healthy behavior and ideal weight," *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI*, 2011.
- [78] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury e A. T. Campbell, "The Jigsaw Continuous Sensing Engine for Mobile Phone Applications," *Proceedings of the 8th Conference on Embedded Networked Sensor Systems (SenSys'10)*, pp. 71-84, 2010.
- [79] S. Zhang, P. McCullagh, C. Nugent e H. Zheng, "Activity Monitoring Using a Smart Phone's Accelerometer with Hierarchical Classification," *Intelligent Environments (IE), Sixth International Conference on IEEE*, pp. 158-163, 2010.
- [80] A. Henpraserttae, S. Thiemjarus e S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," *Proceedings - 2011 International Conference on Body Sensor Networks, BSN 2011*, pp. 41-46, 2011.
- [81] J. R. Kwapisz, G. M. Weiss e S. A. Moore, "Activity Recognition using Cell Phone Accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, p. 74, 2011.
- [82] B. Das, A. M. Seelye, B. L. Thomas, D. J. Cook, L. B. Holder e M. Schmitter-Edgecombe, "Using smart phones for context-aware prompting in smart environments," *IEEE Consumer Communications and Networking Conference, CCNC'2012*, pp. 399-403, 2012.
- [83] J. B. Gomes, S. Krishnaswamy, M. M. Gaber, P. a. C. Sousa e E. Menasalvas, "MARS: A personalised mobile activity recognition system," *Proceedings - 2012 IEEE 13th International Conference on Mobile Data Management, MDM 2012*, pp. 316-319, 2012.

- [84] M. Kose, O. D. Incel e C. Ersoy, "Online Human Activity Recognition on Smart Phones," *2nd International Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*, 2012.
- [85] J. J. Guiry, P. van de Ven e J. Nelson, "Multi-sensor fusion for enhanced contextual awareness of everyday activities with ubiquitous devices," *Sensors*, vol. 14, n° 3, pp. 5687-5701, 2014.
- [86] A. M. Khan, M. H. Siddiqi e S. W. Lee, "Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones," *Sensors*, vol. 13, n° 10, pp. 13099-13122, 2013.
- [87] O. Banos, J.-M. Galvez, M. Damas, H. Pomares e I. Rojas, "Window Size Impact in Human Activity Recognition," *Sensors*, vol. 14, n° 4, pp. 6474-6499, 2014.
- [88] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin e H.-P. Tan, "Deep Activity Recognition Models with Triaxial Accelerometers," *Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 2015.
- [89] Z. S. Abdallah, M. M. Gaber, B. Srinivasan e S. Krishnaswamy, "AnyNovel: detection of novel concepts in evolving data streams," *Evolving Systems*, pp. 1-21, 2016.
- [90] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra e D. Anguita, "Transition-Aware Human Activity Recognition Using Smartphones," *Neurocomputing*, vol. 171, pp. 754-767, 2016.
- [91] M. Shoaib, S. Bosch, O. Durmaz Incel, H. Scholten e P. J. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors (Switzerland)*, vol. 14, n° 6, pp. 10146-10176, 2014.
- [92] W. Sousa, E. Souto, J. Rodrigues, P. Sadarc, R. Jalali e K. El-khatib, "A Comparative Analysis of the Impact of Features on Human Activity Recognition with Smartphone Sensors," *Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web*, pp. 397-404, 2017.
- [93] A. Wang, G. Chen, J. Yang, S. Zhao e C. Y. Chang, "A Comparative Study on Human Activity Recognition Using Inertial Sensors in a Smartphone," *IEEE Sensors Journal*, 2016.
- [94] B. Almaslukh, J. Almuhtadi e A. Artoli, "An Effective Deep Autoencoder Approach for Online Smartphone- Based Human Activity Recognition," *IJCSNS International Journal of Computer Science and Network Security*, vol. 17, n° 4, pp. 160-165, 2017.
- [95] L. Köping, K. Shirahama e M. Grzegorzec, "A General Framework for Sensor-based Human Activity Recognition," *Computers in Biology and Medicine*, 2018.
- [96] A. Sucerquia, J. D. López e J. F. Vargas-Bonilla, "Real-Life/Real-Time Elderly Fall Detection with a Triaxial Accelerometer," *Sensors*, vol. 18, n° 4, 2018.
- [97] M. F. A. b. Abdullah, F. P. N. Ali, M. S. Sayeed, D.-J. Choi e K. S. Muthu, "Classification algorithms in human activity recognition usign smartphones," *International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering*, vol. 6, n° 8, pp. 362-369, 2012.

- [98] A. Bulling, U. Blanke e B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys*, vol. 46, nº 3, pp. 1-33, 2014.
- [99] M. Ehatisham-ul-Haq, M. A. Azam, J. Loo, K. Shuang, S. Islam, U. Naeem e Y. Amin, "Authentication of Smartphone Users Based on Activity Recognition and Mobile Sensing," *Sensors*, vol. 17, nº 9, p. 2043, 2017.
- [100] C. Shen, Y. Chen e X. Guan, "Performance evaluation of implicit smartphones authentication via sensor-behavior analysis," *Information Sciences*, pp. 538-553, 2018.
- [101] M. Shoaib, S. Bosch, O. Incel, H. Scholten e P. Havinga, "A Survey of Online Activity Recognition Using Mobile Phones," *Sensors*, vol. 15, 2015.
- [102] Y. Vaizman, K. Ellis e G. Lanckriet, "Recognizing Detailed Human Context In-the-Wild from Smartphones and Smartwatches," *IEEE Pervasive Computing*, vol. 16, nº 4, pp. 62-74, 2017.
- [103] J. Wang, Y. Chen, S. Hao, X. Peng e L. Hu, "Deep Learning for Sensor-based Activity Recognition: A Survey," *Computer Vision and Pattern Recognition*, 2017.
- [104] E. M. Tapia, "Using machine learning for real-time activity recognition and estimation of energy expenditure," *MIT, PhD Thesis*, 2008.
- [105] A. Avci e S. Bosch, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," *Architecture of computing systems (ARCS)*, pp. 1-10, 2010.
- [106] D. Peebles, H. Lu, N. D. Lane, T. Choudhury e A. T. Campbell, "Community-Guided Learning : Exploiting Mobile Sensor Users to Model Human Behavior," *Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 10, pp. 1600-1606, 2008.
- [107] X. Bao, J. Yang, Z. Yan, L. Luo, Y. Jiang, E. M. Tapia e E. Welbourne, "CommSense : Identify Social Relationship with Phone Contacts via Mining Communications," *Mobile Data Management (MDM)*, pp. 227-234, 2015.
- [108] Z. Yan, J. Yang e E. Tapia, "Smartphone bluetooth based social sensing," *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pp. 95-98, 2013.
- [109] M. Shoaib, H. Scholten e P. Havinga, "Towards Physical Activity Recognition Using Smartphone Sensors," *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, vol. 19, pp. 80-87, 2013.
- [110] M. Mladenov e M. Mock, "A step counter service for Java-enabled devices using a built-in accelerometer," *Proceedings of the 1st International Workshop on Context-Aware Middleware and Services affiliated with the 4th International Conference on Communication System Software and Middleware (COMSWARE 2009) - CAMS '09*, 2009.
- [111] J. W. Lockhart e G. M. Weiss, "Limitations with Activity Recognition Methodology & Data Sets," *2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing:*

*Adjunct Publication*, pp. 747-756, 2014.

- [112] N. Lane, M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury e A. Campbell, "BeWell: A Smartphone Application to Monitor, Model and Promote Wellbeing," *Proceedings of the 5th International ICST Conference on Pervasive Computing Technologies for Healthcare*, p. 8, 2011.
- [113] O. D. Lara e M. a. Labrador, "A mobile platform for real-time human activity recognition," *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pp. 667-671, 2012.
- [114] P. Siirtola e J. Rönning, "Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 1, nº 5, p. 38, 2012.
- [115] J.-g. Park, A. Patel, D. Curtis, S. Teller e J. Ledlie, "Online pose classification and walking speed estimation using handheld devices," *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, 2012.
- [116] Z. Yan, A. Misra, D. Chakraborty, K. Aberer e H. Jeung, "Semantic activity classification using locomotive signatures from mobile phones," *EPFL-REPORT*, pp. 1-9, 2012.
- [117] Y. Ustev, O. D. Incel e C. Ersoy, "User, device and orientation independent human activity recognition on mobile phones: challenges and a proposal," *ACM conference on Pervasive and ubiquitous computing adjunct publication*, 2013.
- [118] A. Anjum e M. U. Ilyas, "Activity recognition using smartphone sensors," *IEEE 10th Consumer Communications and Networking Conference, CCNC 2013*, pp. 914-919, 2013 .
- [119] P. Siirtola e J. Rönning, "Ready to use activity recognition for smartphones," *IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013*, pp. 59-64, 2013.
- [120] S. Thiemjarus, A. Henpraserttae e S. Marukatat, "A study on instance-based learning with reduced training prototypes for device-context-independent activity recognition on a mobile phone," *2013 IEEE International Conference on Body Sensor Networks, BSN 2013*, 2013.
- [121] M. Hynes, H. Wang, E. McCarrick e L. Kilmartin, "Accurate monitoring of human physical activity levels for medical diagnosis and monitoring using off-the-shelf cellular handsets," *Personal and Ubiquitous Computing*, vol. 15, nº 7, pp. 667-678, 2011.
- [122] C. K. Schindhelm, "Activity recognition and step detection with smartphones: Towards terminal based indoor positioning system," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, pp. 2454-2459, 2012.
- [123] H. Martín, A. M. Bernardos, J. Iglesias e J. R. Casar, "Activity logging using lightweight classification techniques in mobile devices," *Personal and Ubiquitous Computing*, vol. 17, nº 4, pp. 675-695, 2013.
- [124] J. Fontecha, F. J. Navarro, R. Hervás e J. Bravo, "Elderly frailty detection by using accelerometer-enabled smartphones and clinical information records," *Personal and*



*Ubiquitous Computing*, vol. 17, nº 6, pp. 1073-1083, 2013.

- [125] Q. V. Vo, M. T. Hoang e D. Choi, "Personalization in mobile activity recognition system using-medoids clustering algorithm," *International Journal of Distributed Sensor Networks*, 2013.
- [126] S. Ramamurthy e N. Roy, "Recent trends in machine learning for human activity recognition—A survey," *WIRES: Data Mining Knowledge Discovery*, pp. 1-11, 2018.
- [127] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán e D. Roggen, "The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, nº 15, pp. 2033-2042, 2013.
- [128] D. Anguita, A. Ghio, L. Oneto, X. Parra e J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 24-26, 2013.
- [129] A. Reiss, M. Weber e D. Stricker, "Exploring and extending the boundaries of physical activity recognition," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 46-50, 2011.
- [130] M. Zhang e A. A. Sawchuk, "USC-HAD: A Daily Activity Dataset for Ubiquitous Activity Recognition Using Wearable Sensors," *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, p. 1036, 2012.
- [131] O. Banos, R. Garcia, J. A. Holgado, M. Damas, H. Pomares, I. S. Rojas e V. C. A., "mHealthDroid: a novel framework for agile development of mobile health applications," *Proceedings of the 6th International Work-conference on Ambient Assisted Living an Active Ageing (IWAAL 2014)*, 2014.
- [132] D. Micucci, M. Mobilio e P. Napolitano, "UniMiB SHAR: a new dataset for human activity recognition using acceleration data from smartphones," *Computer Vision and Pattern Recognition*, 2017.
- [133] D. J. Cook e N. C. Krishnan, *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*, John Wiley & Sons, 2015.
- [134] J. Lin, E. Keogh, L. Wei e Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, pp. 107-144, 2007.
- [135] R. Sebastião, M. M. Silva, R. Rabiço, J. Gama e T. Mendonça, "Real-time algorithm for changes detection in depth of anesthesia signals," *Evolving Systems*, vol. 4, nº 1, pp. 3-12, 2013.
- [136] J. W. Lockhart, "The Benefits of Personalized Data Mining Approaches to Human Activity Recognition with Smartphone Sensor Data," *Masters Thesis*, p. 46, 2014.
- [137] D. C. Lay, "Linear algebra and its applications," pp. 14-25, 2005.
- [138] G. Strang, "Wavelets," *Sigma Xi, The Scientific Research Society*, vol. 82, pp. 250-255, 1994.

- [139] D. Chu, N. D. Lane, T. T.-T. Lai, C. Pang, X. Meng, Q. Guo, F. Li e F. Zhao, "Balancing energy, latency and accuracy for mobile sensor data classification," *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems - SenSys '11*, p. 54, 2011.
- [140] J. Lee e M. and Verleysen, *Nonlinear dimensionality reduction*, Springer Science & Business Media, 2007.
- [141] A. M. Khan, "Human Activity Recognition Using A Single Tri-axial Accelerometer," *PhD Thesis*, p. 1379–1383, 2011.
- [142] M. M. Hassan, M. Z. Uddin, A. Mohamed e A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, 2017.
- [143] M. H. Siddiqi, R. A. Ali, M. S. Rana, E.-K. Hong, E. S. Kim e S. Lee, "Video-Based Human Activity Recognition Using Multilevel Wavelet Decomposition and Stepwise Linear Discriminant Analysis," *Sensors*, vol. 14, 2014.
- [144] M. Guo e Z. Wang, "Segmentation and recognition of human motion sequences using wearable inertial sensors," *Multimedia Tools and Applications*, pp. 1-20, 2018.
- [145] C. E. Garcia e R. F. Brena, "Activity recognition using community data to complement small amounts of labeled instances," *Sensors*, vol. 16, nº 6, 2016.
- [146] Y. Bengio, "Deep learning of representations: Looking forward," *International Conference on Statistical Language and Speech Processing*, p. 1–37, 2013.
- [147] Y. LeCun, Y. Bengio e G. Hinton, "Deep learning," *Nature*, p. 436–444, 2015.
- [148] Y. Li, D. Shi, B. Ding e D. Liu, "Unsupervised feature learning for human activity recognition using smartphone sensors," *Mining Intelligence and Knowledge Exploration*, p. 99–107, 2014.
- [149] Y. Zheng, Q. Liu, E. Chen, Y. Ge e J. Zhao, "Time series classification using multi-channels deep convolutional neural networks," *International Conference on Web-Age Information Management*, p. 298–310, 2014.
- [150] J. Wang, Y. Chen, S. Hao, X. Peng e L. Hu, "Deep Activity Recognition Models with Triaxial Accelerometers," *Computer Vision and Pattern Recognition*, 2017.
- [151] S. Bhattacharya e N. Lane, "From smart to deep: Robust activity recognition on smartwatches using deep learning," *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (Per-*, p. 1–6, 2016.
- [152] N. Hammerla, J. Fisher, P. Andras, L. Rochester e R. Walker, "Pd disease state assessment in naturalistic environments using deep learning," *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [153] T. Hayashi, M. Nishida, N. Kitaoka e K. Takeda, "Daily activity recognition based on dnn using environmental sound and acceleration signals," *Signal Processing Conference (EUSIPCO)*, p. 2306–2310, 2015.

- [154] N. Lane e P. Georgiev, "Can deep learning revolutionize mobile sensing?," *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, p. 117–122, 2015.
- [155] C. Liu, L. Zhang, Z. Liu, K. Liu, X. Li e Y. Liu, "Lasagna: towards deep hierarchical understanding and searching over mobile sensing data," *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, p. 334–347, 2016.
- [156] P. T., N. Hammerla e P. Olivier, "Feature learning for activity recognition in ubiquitous computing," *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, p. 1729, 2011.
- [157] V. Radu, N. Lane, S. Bhattacharya, C. Mascolo, M. Marina e F. Kawsar, "Towards multimodal deep learning for activity recognition on mobile devices," *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, p. 185–188, 2016.
- [158] L. Zhang, X. Wu e D. Luo, "Real-time activity recognition on smart-phones using deep neural networks," *Ubiquitous Intelligence and Computing (UIC)*, p. 1236–1242, 2015.
- [159] Y. Chen e Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," *Systems, Man, and Cybernetics (SMC)*, p. 1488–1492, 2015.
- [160] Y. Chen, K. Zhong, J. Zhang, Q. Sun e X. Zhao, "Lstm networks for mobile human activity recognition".
- [161] H. Gjoreski, J. Bizjak, M. Gjoreski e M. Gams, "Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer," *IJCAI-16 workshop on Deep Learning for Artificial Intelligence (DLAI)*, 2016.
- [162] S. Ha, J. Yun e S. Choi, "Multi-modal convolutional neural networks for activity recognition," *Systems, Man, and Cybernetics (SMC)*, p. 3017–3022, 2015.
- [163] S. Ha e S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," *Neural Networks (IJCNN)*, p. 381–388, 2016.
- [164] N. Hammerla, S. Halloran e T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *IJCAI*, 2016.
- [165] J. Hannink, T. Kautz, C. Pasluosta, K. Gabmann, J. Klucken e B. Eskofier, "Sensor-based gait parameter extraction with deep convolutional neural networks," *IEEE journal of biomedical and health informatics*, p. 85–93, 2017.
- [166] W. Jiang e Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," *Proceedings of the 23rd ACM international conference on Multimedia*, p. 1307–1310, 2015.
- [167] Y. Kim e Y. Li, "Human activity classification with transmission and reflection coefficients of on-body antennas through deep convolutional neural networks," *IEEE Transactions on Antennas and Propagation* 65, p. 2764–2768, 2017.

- [168] S. Lee, S. Yoon e H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," *Big Data and Smart Computing (BigComp)*, p. 131–134, 2017.
- [169] S. Mohammed e I. Tashev, "Unsupervised deep representation learning to remove motion artifacts in free-mode body sensor networks," *Wearable and Implantable Body Sensor Networks (BSN)*, p. 183–188, 2017.
- [170] F. Morales e D. Roggen, "Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations," *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, p. 92–99, 2016.
- [171] B. Pourbabaee, M. Roshtkhari e K. Khorasani, "Deep convolution neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.
- [172] D. Ravi, C. Wong, B. Lo e G. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," *Wearable and Implantable Body Sensor Networks (BSN)*, p. 71–76, 2016.
- [173] D. Ravi, C. Wong, B. Lo e G. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE journal of biomedical and health informatics*, p. 56–64, 2017.
- [174] C. Ronao, S. Cho e 2015, "Deep convolutional neural networks for human activity recognition with smartphone sensors," *International Conference on Neural Information Processing*, p. 46–53, 2015.
- [175] A. Sathyanarayana, S. Joty, L. Fernandez-Luque, F. Ofli, J. Srivastava, A. Elmagarmid, S. Taheri e T. Arora, "Impact of physical activity on sleep: A deep learning based exploration," *arXiv preprint:1607.07034*, 2016.
- [176] J. Wang, X. Zhang, Q. Gao, H. Yue e H. Wang, "Device-free wireless localization and activity recognition: A deep learning approach," *IEEE Transactions on Vehicular Technology*, 2016.
- [177] J. Yang, M. Nguyen, P. San, X. Li e S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, p. 25–31, 2015.
- [178] S. Yao, S. Hu, Y. Zhao, A. Zhang e T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," *Proceedings of the 26th International Conference on World Wide Web, International WWW Conferences Steering Committee*, p. 351–360, 2017.
- [179] T. Zebin, P. Scully e K. Ozanyan, "Human activity recognition with inertial sensors using a deep learning approach," *SENSORS*, p. 1–3, 2016.
- [180] M. Zeng, L. Nguyen, B. Yu, O. Mengshoel, J. Zhu, P. Wu e J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," *Mobile Computing, Applications and Services (Mobi-)*, p. 197–205, 2014.

- [181] Y. Zheng, Q. Liu, E. Chen, Y. Ge e J. Zhao, "Exploiting multi- channels deep convolutional neural networks for multivariate time series classificatio," *Frontiers of Computer Science* 10,, p. 96–112, 2016.
- [182] M. Edel e Koppe, "Binarized-blstm-rnn based human activity recognition," *Indoor Positioning and Indoor Navigation (IPIN)*, p. 1–7, 2016.
- [183] Y. Guan e T. Ploetz, "Ensembles of deep lstm learners for activity recog- nition using wearables," *arXiv:1703.09370*, 2017.
- [184] M. Inoue, S. Inoue e T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput," *arXiv preprint arXiv:1611.03607*, 2016.
- [185] F. Ordóñez e D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, 2016.
- [186] P. Vepakomma, D. De, S. Das e S. Bhansali, "A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities," *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 1-6, 2015.
- [187] K. Walse, R. Dharaskar e V. Thakare, "Pca based optimal ann classifiers for human activity recognition using mobile sensors data," *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems*, vol. 1, p. 429–436, 2016.
- [188] L. Zhang, X. Wu e D. Luo, "Human activity recognition with hmm-dnn model," *Cognitive Informatics&Cognitive Computing (ICCI\* CC)*, p. 192–197, 2015.
- [189] L. Zhang, X. Wu e D. Luo, "Recognizing human activities from raw accelerometer data using deep neural networks," *2015 IEEE 14th Interna- tional Conference on Machine Learning and Applications (ICMLA)*, p. 865–870, 2015.
- [190] Z. Chen, L. Zhang, Z. Cao e J. Guo, "Distilling the Knowledge from Handcrafted Features for Human Activity Recognition," *IEEE Transactions on Industrial Informatics*, vol. 3203, pp. 1-9, 2018.
- [191] L. A. Klein, "Sensor and data fusion: a tool for information assessment and decision making," *Bellingham eWA WA: Spie Press*, 2004.
- [192] P. Tsinganos e A. Skodras, "On the Comparison of Wearable Sensor Data Fusion to a Single Sensor Machine Learning Technique in Fall Detection," *Sensors*, vol. 18, nº 2, p. 592, 2018.
- [193] J. B. Bancroft e G. Lachapelle, "Data Fusion Algorithms for Multiple Inertial Measurement Units," *Sensors*, vol. 11, pp. 6771-6798, 2011.
- [194] P. Schäfer, "Scalable Time Series Similarity Search for Data Analytics," *Dissertation*, p. 175, 2015.
- [195] Y. Cai e R. Ng, "Indexing spatio-temporal trajectories with Chebyshev polynomials," *ACM SIGMOD international conference on Management of data - SIGMOD '04*, p. 599, 2004.

- [196] Q. Chen, L. Chen, X. Lian, Y. Liu e J. X. Yu, "Indexable PLA for efficient similarity search," *International conference on Very large data bases*, pp. 435-446, 2007.
- [197] E. J. Keogh e M. J. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," *Knowledge Discovery and Data Mining*, vol. 1805, pp. 122-133, 2000.
- [198] B. Yi e C. Faloutsos, "Fast time sequence indexing for arbitrary  $l_p$  norms," *The International Journal on Very Large Data Bases*, p. 385-394, 2000.
- [199] J. Lin, E. Keogh, S. Lonardi e B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," *SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 2-11, 2003.
- [200] A. Camera, T. Palpanas, J. Shieh e E. Keogh, "iSAX 2.0: Indexing and mining one billion time series," *IEEE International Conference on Data Mining, ICDM*, pp. 58-67, 2010.
- [201] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen e S. Frankenstein, "Time series anomaly discovery with grammar-based compression," *EDBT*, pp. 481-492, 2015.
- [202] S. M. Wong, W. Ziarko e P. C. Wong, "Generalized vector spaces model in information retrieval," *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, 1985.
- [203] R. Lowry, "Concepts and applications of inferential statistics," *Book*, 2014.
- [204] P. Schäfer e U. Leser, "Multivariate Time Series Classification with Temporal Abstractions," *arXiv preprint arXiv:1711.11343*, 2017.
- [205] M. G. Baydogan e G. Runger, "Learning a symbolic representation for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 29, n° 2, p. 400-422, 2015.
- [206] K. G. M. Quispe, W. S. Lima e E. J. P. Souto, "Human Activity Recognition on Smartphones using Symbolic Data Representation," em *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, Salvador, BA, Brazil, 2018 .
- [207] E. Keogh, J. Lin e A. Fu, "HOT SAX: Finding the most Unusual Time Series Subsequences: Algorithms and Application," *International Conference on Data Mining*, 2005.
- [208] M. Terzi, A. Cenedese e G. Susto, "A multivariate symbolic approach to activity recognition for wearable applications," *IFAC-PapersOnLine*, pp. 15865-15870, 2017.
- [209] W. H. Gomaa e A. A. Fahmy, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications*, vol. 68, n° 13, pp. 13-18, 2013.
- [210] B. D. e J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," *Conference on Artificial Intelligence*, pp. 229-248, 1994.
- [211] T. Miu, P. Missier e T. Plotz, "Bootstrapping Personalised Human Activity Recognition Models Using Online Active Learning," *EEE International Conference on Computer and*

*Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 1138-1147, 2015.

- [212] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy e A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, n° 4, pp. 1-37, 2014.
- [213] G. a. D. P. Hulten, "Mining complex models from arbitrarily large databases in constant time," *International Conference on Knowledge Discovery and Data Mining*, p. 525–531, 2002.
- [214] N. C. O. a. S. Russell, "Online bagging and boosting," *Artificial Intelligence and Statistics*, p. 105–112, 2001.
- [215] R. Kohavi e C. Kunz, "Option decision trees with majority votes," *International Conference on Machine Learning*, p. 161–169, 1997.
- [216] A. V. Oppenheim e R. W. Schaffer, *Digital Signal Processing*, Prentice Hall, 1975.
- [217] Sung-Hyuk Cha e S. N. Srihari, "On measuring the distance between histograms," *Pattern Recognition*, vol. 35, n° 6, pp. 1355-1370, 2002.
- [218] R. Klinkenberg, "Learning drifting concepts: Example selection vs. example weighting," *Intelligent Data Analysis*, vol. 8, n° 3, p. 281–300, 2004.
- [219] D. Anguita, A. Ghio, L. Oneto, X. Parra e J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, pp. 24-26, 2013.
- [220] M. Lohrey, S. Maneth e R. Mennicke, "Tree Structure Compression with RePair," *Data Compression Conference (DCC)*, 2011.
- [221] C. G. Nevill-Manning e I. H. Witten, "Identifying Hierarchical Structure in Sequences: A linear-time algorithm," *arXiv preprint cs/9709102*, vol. 7, pp. 67-82, 1997.
- [222] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand e H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [223] K. Ouchi e M. Doi, "Indoor-outdoor activity recognition by a smartphone," *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, p. 600, 2012.
- [224] P. B. Romera, M. S. Aung e N. Bianchi-Berthouze, "A one-vs-one classifier ensemble with majority voting for activity recognition," *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 443-448, 2013.

# APÊNDICE A.

## BASES DE DADOS

Este apêndice descreve detalhes das bases de dados utilizadas nos experimentos deste trabalho. Todas elas são comumente utilizadas na literatura de HAR baseada em smartphones instrumentadas com sensores inerciais.

### 1.1. UCI-HAR

A base de dados UCI-HAR<sup>5</sup> [219] foi construída com dados coletados de 30 pessoas com idades que variam entre 19 e 48 anos. Cada pessoa executou 6 atividades físicas como andar, sentar, ficar em pé, deitar, subir e descer escadas. Os dados foram coletados de um celular *Samsung Galaxy S2* utilizando os sensores acelerômetro e giroscópio a uma frequência de 50 Hz. A coleta foi realizada com o celular localizado na cintura dos usuários. Todas as etapas de coleta de dados foram gravadas<sup>6</sup> e os dados foram rotulados manualmente.

A primeira versão da base de dados UCI-HAR possui dois tipos de dados. O primeiro consiste nos dados brutos, extraídos do sensor acelerômetro do smartphone. O segundo consiste nos dados processados com as seguintes configurações: a) as informações referentes à força da gravidade foram retiradas com o propósito de manter somente informações do movimento corporal dos usuários; b) o filtro *Butterworth Low-pass* com uma frequência de corte de 0.3 Hz foi utilizado na tentativa de eliminar o ruído do sinal; c) os dados foram segmentados com uma janela de tempo de 2.56 segundos e uma taxa de sobreposição de 50% e, d) a base de dados possui 561 características incluindo os domínios de tempo e frequência. A UCI-HAR possui 1.311.439 amostras.

Recentemente, uma nova versão da base de dados UCI-HAR, denominada de HAPT (*Human Activities and Postural Transitions*) foi disponibilizada em <sup>7</sup>. Essa base de dados contém dados dos sensores acelerômetro e giroscópio e, também, informações de transição de atividades como de sentar e levantar (vice-versa), sentar e deitar (vice-versa) e deitar e levantar (vice-versa). As informações relacionadas à coleta de dados, frequência e posição do celular no corpo do usuário são as mesmas da UCI-HAR. Em resumo, as versões das bases de dados utilizada nos experimentos realizados nesta tese possuem as seguintes configurações:

1. **UCI-HAR-A (UA)** - composta por dados do acelerômetro com informações de gravidade;
2. **UCI-HAR-LA (ULA)** - composta por dados do acelerômetro sem informação de gravidade (acelerômetro linear);
3. **HAPT (HAG)** - composta por dados de acelerômetro e giroscópio. Os dados do acelerômetro possuem informações de gravidade;
4. **HAPT-G (HG)** – composta por dados do giroscópio;
5. **HAPT-A (HA)** - composta por dados do acelerômetro com informações de gravidade;

<sup>5</sup> <https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

<sup>6</sup> [https://www.youtube.com/watch?v=XOEN9W05\\_4A](https://www.youtube.com/watch?v=XOEN9W05_4A)

<sup>7</sup> <http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions>



A Figura A-1 mostra um gráfico com a distribuição dos dados referente as atividades que compõem a base de dados UCI-HAR.

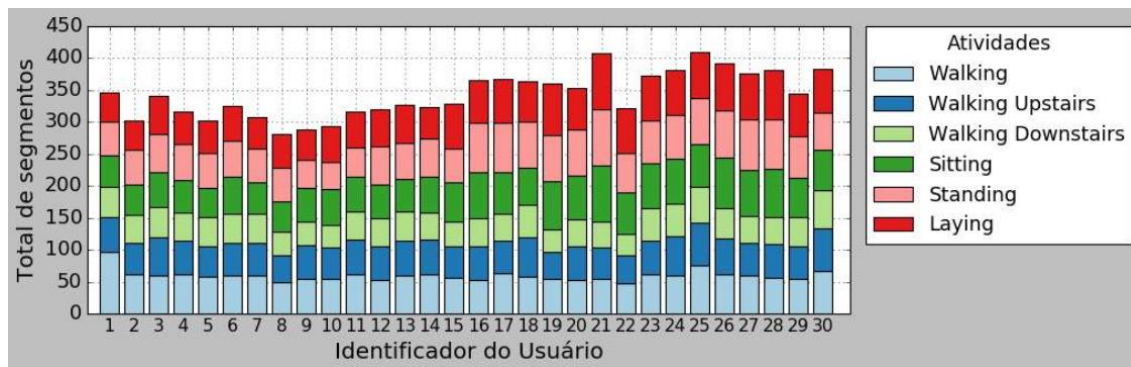


Figura A-1. Exemplo da distribuição das classes por atividade na base de dados UCI-HAR.

## 1.2. Shoaib

A base de dados Shoaib [91] foi construída com dados de 10 usuários do sexo masculino com idades variando de 25 e 30 anos. Cada usuário executou 8 atividades no período de 3 a 4 minutos para cada uma. Tais atividades incluem andar, correr, sentar, ficar de pé, correr lentamente, andar de bicicleta, subir escadas e descer escadas. Os dados foram coletados de um celular *Samsung Galaxy S2 (i9100)* utilizando os sensores acelerômetro, acelerômetro linear, giroscópio e magnetômetro a uma frequência de 50 Hz. Os usuários foram equipados com cinco smartphones localizados em cinco posições do corpo incluindo bolso direito e esquerdo da calça, cintura, pulso e antebraço, conforme mostrado na Figura A-2. A Shoaib possui 629.977 amostras.

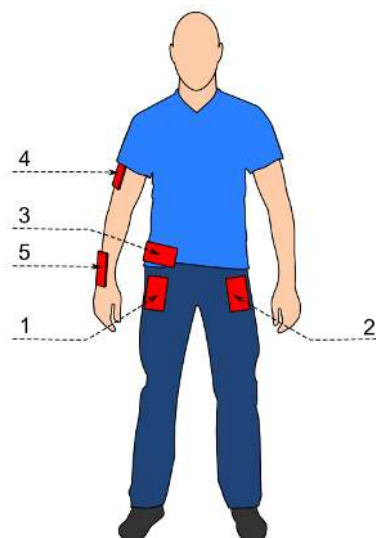


Figura A-2. Posições do smartphone no corpo do usuário de onde os dados da base Shoaib foram coletados. Fonte [91].

## 1.3 WISDM

A base de dados *Wireless Sensor Data Mining (WISDM)* [81] foi construída com dados de 36 usuários. Cada usuário executou 6 atividades como andar, sentar, ficar em pé, deitar, subir e

descer escadas. Os dados foram coletados de um smartphone com Android (*Nexus One*, *HTC Hero*) a uma frequência de 20Hz. A coleta foi realizada com o celular localizado no bolso frontal da calça do usuário. A Figura A-3 mostra um gráfico com a distribuição dos dados referente às atividades que compõem a base de dados WISDM. A WISDM possui 1.098.830 amostras.

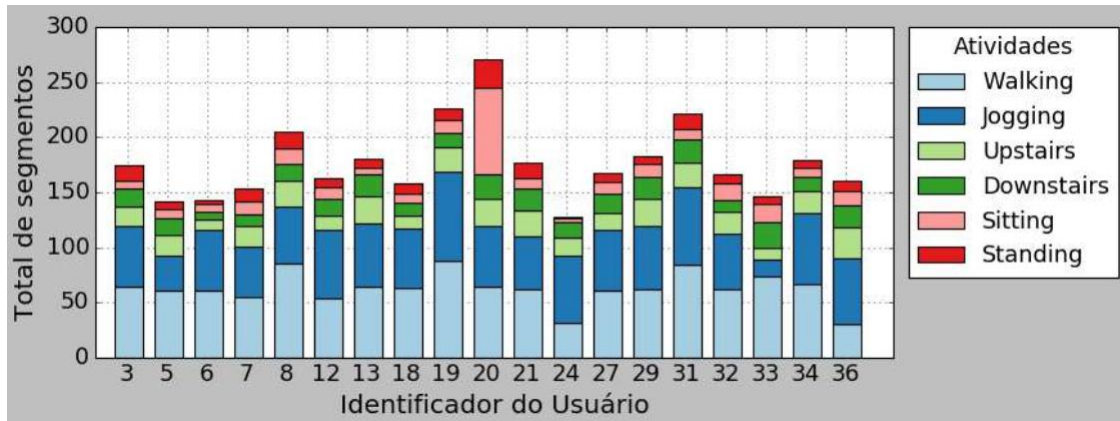


Figura A-3. Exemplo da distribuição das classes por atividade na base de dados WISDM.

# APÊNDICE B. MÉTODOS E CARACTERÍSTICAS USADOS EM SISTEMAS HAR

Este apêndice apresenta duas tabelas com mais detalhes relacionados a distribuição dos trabalhos por extração de características e algoritmos de classificação.

TABLE I. LISTA DE TRABALHOS SEPARADOS POR CADA CARACTERÍSTICAS.

<b>Características</b>	<b>Trabalhos</b>
Magnitude	[74] [29] [110] [76] [63] [57] [121] [84] [113] [114] [115] [116] [117] [118] [119]
Horizontal	[29] [122] [115] [116]
Vertical	[29] [80] [122] [115] [116] [120]
FFT	[74] [67] [76] [63] [57] [122] [115] [116] [117] [118] [123]
Wavelets	[73]
Mean	[72] [30] [74] [67] [29] [110] [76] [31] [78] [63] [57] [64] [81] [112] [53] [82] [84] [113] [223] [122] [114] [115] [116] [117] [118] [123] [119] [120] [125] [124] [124] [54]
Min	[72] [74] [63] [53] [83] [84] [223] [114] [119]
Max	[72] [74] [63] [53] [83] [84] [223] [114] [119]
Variance	[72] [76] [31] [78] [57] [64] [112] [53] [113] [223] [116] [117] [118] [123] [124]
Standard Deviation	[74] [67] [29] [81] [84] [113] [122] [114] [117] [118] [119] [120] [125] [124] [54]
Peaks	[67] [110] [81] [122] [118] [125]
Amplitude	[124]
Peaks Amplitude	[82]
Zero Crossing Rate	[29] [53] [117] [123]
Mean Crossing Rate	[78] [112]
Quartil	[29]
Percentil	[114] [123] [119]
Interquartil	[29] [113] [123]
Correlation	[29] [53] [116] [117] [123] [54]
Cross-correlation	[118]
Auto-correlation	[118] [86]
Absolute Mean Deviation	[81] [113] [124]
Binned Distribution	[81]
AMDF	[121]
Root Mean Square	[82] [113]
Norma Euclidiana	[83]
Sum	[119]
Square Sum	[119]
Sum Square Error	[113]
Signal Magnitude Area	[116] [54]
Signal Magnitude	[116]

Mean	
Pearson Coefficient	[124]
Entropy	[73] [29] [78] [112] [116] [118] [123]
Energy	[74] [78] [57] [112] [113] [122] [116] [118] [123] [125]
Centroid	[29] [123]
DC Component	[115] [118]
Tilt Angle	[115]

TABLE II. LISTA DE TRABALHOS SEPARADOS POR MÉTODOS.

<b>Método</b>	<b>Trabalhos</b>
Naïve bayes	[72] [73] [74] [29] [78] [63] [53] [82] [83] [84] [116] [118] [123]
Decision Tree Family	[67] [29] [75] [76] [78] [57] [81] [53] [82] [113] [122] [116] [118] [123] [119]
SVM	[72] [29] [79] [78] [82] [223] [115] [116] [118] [125] [54]
KNN	[29] [75] [63] [80] [82] [84] [114] [117] [118] [120]
Redes Neurais	[30] [81] [53] [86]
Logistic Regression	[81]
Regression	[86]
Boosting	[72]
AdaBoost	[116]
HMM	[30] [75]
DHMM	[57]
GMM	[75]
Fuzzy	[31]
Multivariate Gaussian Model	[78]
k-star	[53]
Quadratic Discriminant Analysis	[114] [119]
RLS (Regularized least square)	[115]
Kernel Discriminant Analysis	[86] [54]
K-Means	[30]
Self-organized map	[73]
k-metoids clustering	[125]

# APÊNDICE C. AVALIAÇÃO DOS ALGORITMOS OFFLINE

Este apêndice trata da avaliação dos algoritmos baseados na aprendizagem *offline* mapeados na literatura. Os algoritmos são avaliados com base nas métricas descritas no Capítulo 4 e, portanto, as avaliações se concentram nos resultados dos modelos de classificação. Os experimentos foram divididos em vários cenários de testes, sendo que cada um é responsável por avaliar técnicas e configurações específicas como: 1) Tamanho de janelas de tempo e sobreposição; 2) domínio de características; 3) técnicas de redução de ruídos; 4) fusão de dados; e 5) posição e orientação do smartphone no corpo do usuário. Todas essas técnicas são estudadas com intuito de medir o impacto de cada uma delas na acurácia dos modelos de classificação de atividades gerados pelos algoritmos de classificação.

## 1.1. Protocolo Experimental

Esta seção apresenta uma visão geral de como os experimentos foram planejados, organizados e executados. Dessa forma, os experimentos foram divididos em cinco grupos:

1. O primeiro grupo é focado na avaliação dos modelos de classificação baseados nas características do domínio do tempo e da frequência. Esses experimentos incluem análise de técnicas de redução de ruídos, redução de dimensionalidade e variações nos tamanhos da janela de tempo e taxa de sobreposição.
2. O segundo grupo é focado na avaliação da fusão de dados dos sensores inerciais acelerômetro e giroscópio. O objetivo é verificar o impacto da acurácia dos modelos de classificação quando são utilizados dados de múltiplas fontes de sensores.
3. O terceiro grupo é focado no estudo das posições do smartphone no corpo do usuário. O objetivo é detectar as posições mais relevantes para as atividades de movimento.
4. O quarto grupo é focado numa análise comparativa entre os métodos de aprendizagem de máquina rasos e profundos.

Além disso, outros experimentos, realizados por terceiros, são adicionados neste apêndice com intuito de complementar ou retificar os experimentos realizados por esta pesquisa. Para facilitar a apresentação dos resultados, nós utilizamos uma estratégia baseada na eliminação do uso de métodos e técnicas, ao longo dos experimentos, conforme os resultados forem apresentando baixo desempenho. Essa estratégia tem como propósito facilitar a descrição dos resultados e apresentar apenas as informações importantes para cada cenário de testes.

## 1.2. Domínio do Tempo e Frequência

Os experimentos relacionados a avaliação das características do domínio do tempo e da frequência foram organizados seguindo uma sequência lógica baseada na avaliação das principais soluções na literatura. Dessa forma, os experimentos foram divididos em 5 cenários.

1. O primeiro cenário de testes foca na avaliação das características do domínio do tempo e da frequência utilizando somente os dados do sensor acelerômetro. O objetivo é

descobrir a melhor combinação e configuração dessas características capazes de gerar os melhores modelos de classificação.

2. O segundo cenário de testes foca na avaliação do impacto da aplicação dos filtros na redução de ruídos dos sinais dos sensores inercias.
3. O terceiro cenário de testes foca na análise das técnicas de redução de dimensionalidade via métodos de seleção de características (e.g. *Correlation Attribute Evaluation* e *InfoGain*) e métodos estatísticos que usam combinação lineares e não-lineares (e.g. PCA, LDA e KDA).
4. O quarto cenário de testes foca na avaliação do tamanho e sobreposição das janelas de tempo, assim como, na frequência da coleta de dados.
5. O quinto cenário avalia os melhores modelos de classificação definidos nos cenários anteriores, porém, focado em outras bases de dados. O objetivo é comprovar a eficiência dos modelos em outras bases de dados.

Nos experimentos iniciais foi utilizada a base de dados UCI-HAR, descrita no Apêndice A. Essa base de dados é comumente utilizada pelos principais estudos de análises comparativas identificados na literatura de HAR baseada nos sensores inerciais dos smartphones.

### 1.2.1 - 1º Cenário - Características dos domínios de Tempo e Frequência

Esta seção apresenta uma sequência de resultados de um extensivo estudo realizado com as bases de dados UCI-HAR-A (UA) - composta por dados do acelerômetro com informações de gravidade e UCI-HAR -LA (ULA) - composta por dados do acelerômetro sem informação de gravidade (acelerômetro linear. Ambas possuem 1.311.439 amostras coletadas a uma frequência de 50 Hz.

O propósito deste estudo é verificar e validar a influência das características do domínio de tempo e frequência na acurácia dos modelos de classificação dos algoritmos de aprendizagem de máquina rasos. Para alcançar esse objetivo os testes foram divididos em duas etapas baseadas na orientação do smartphone no corpo do usuário. A primeira etapa trata dos experimentos realizados com as características dependentes de orientação e a segunda etapa trata dos experimentos realizados com as características independentes de orientação. As configurações iniciais definidas nos experimentos são:

- **Tamanho da janela:** 5 segundos;
- **Sobreposição:** 0%;
- **Métodos de classificação:** Família das árvores de decisão (J48, *Decision Stump*, *Random Forest*, *Random Tree*), KNN, *Naive Bayes* e SVM. Todos os algoritmos utilizam os parâmetros padrões configurados pela ferramenta Weka;

Para os métodos de aprendizagem de máquina foram utilizadas as configurações padrões, definidas pela API da ferramenta Weka<sup>8</sup>, relacionadas aos parâmetros de entrada de cada algoritmo. A Tabela C.1 mostra um resumo dos testes realizados nesta etapa. Cada teste foi escolhido baseado nas combinações identificadas no estado da arte.

**Tabela C.1** Conjunto de experimentos realizados na primeira etapa.

Testes	Características
1	Tempo

<sup>8</sup> <https://www.cs.waikato.ac.nz/ml/weka/>

2	Frequência
3	Magnitude (Tempo)
4	Vertical e Horizontal (Tempo)
5	FFT (Frequência)
6	Wavelet (Frequência)

### Características Dependentes de Orientação

As características dependentes de orientação são formadas pelo conjunto das características do domínio do tempo e da frequência. As características do domínio de tempo são usadas de forma isolada, enquanto as características do domínio de frequência são combinadas com as características de baixo nível, FFT e Wavelet.

A Figura C-1 mostra uma comparação das acurácias dos modelos de classificação gerados com todas as características do domínio do tempo descritas na Seção 3.2. Como observado, as taxas de acurácias referentes aos modelos gerados com a base de dados ULA foram, em média, 41.99% inferiores em relação aos modelos gerados com a base de dados UA. Observe que a base de dados ULA corresponde aos dados do acelerômetro sem a informação de gravidade, i.e., os dados do acelerômetro linear. Os modelos de classificação que obtiveram maiores taxas de acurácias foram aqueles derivados da família das árvores de decisão. O modelo do algoritmo *Random Forest*, por exemplo, obteve maior acurácia com uma taxa de 90.63% para base de dados UA.

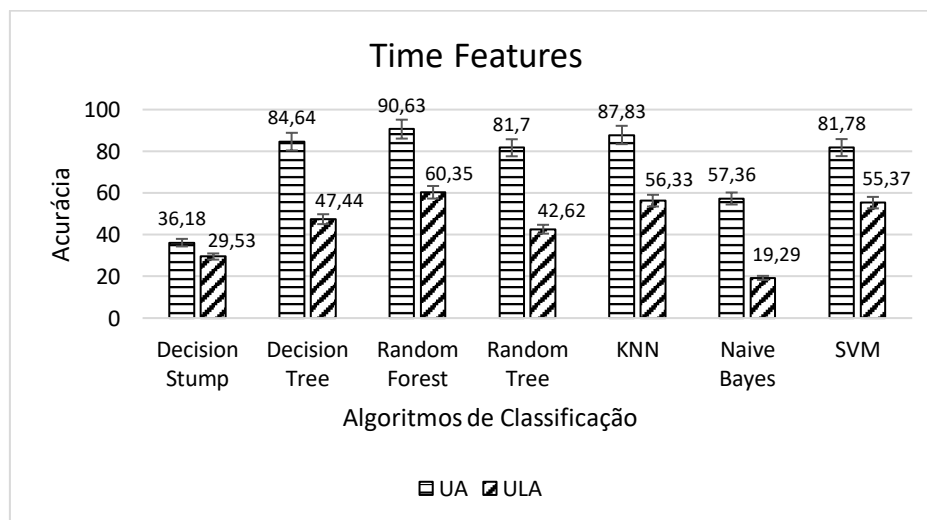
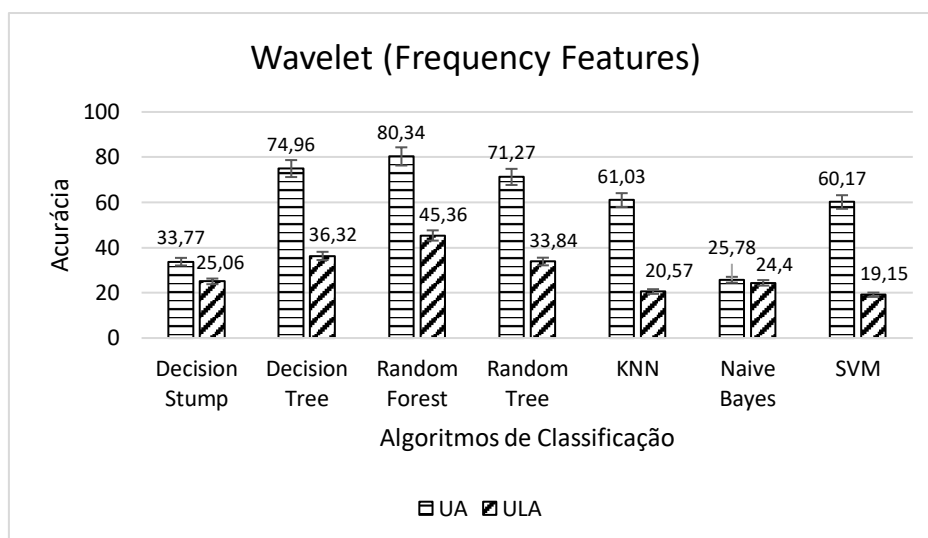
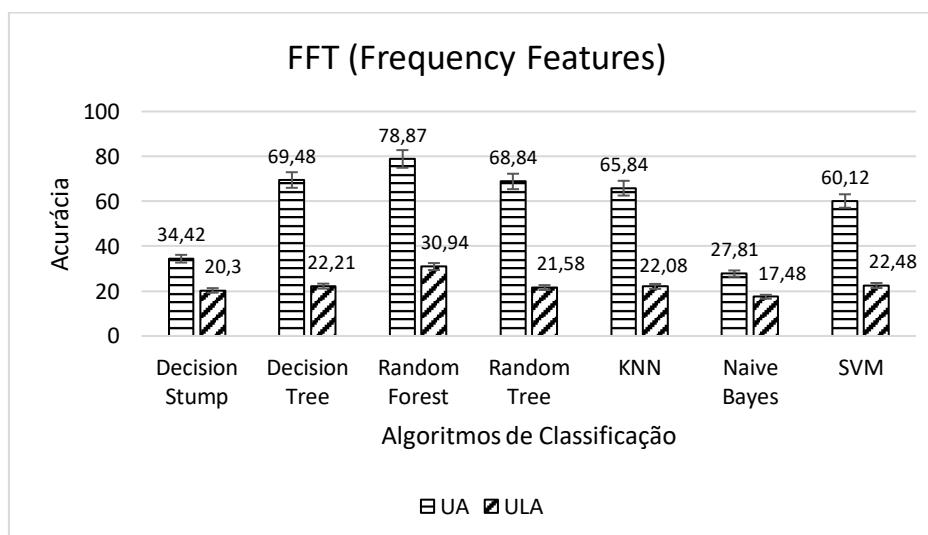


Figura C-1. Acurácia dos modelos de classificação baseado nas características de tempo. Fonte: autor.

A Figura C-2 e a Figura C-3 mostram as taxas de acurácias dos modelos de classificação referente às características do domínio de frequência derivadas das características de baixo nível FFT e *Wavelet*. Os resultados mostram que o uso desses dois conjuntos de características mantém o mesmo padrão dos resultados apresentados na Figura C-1 no sentido de que a base de dados ULA possui resultados inferiores comparados à base de dados UA. Neste caso, algumas diferenças podem ser observadas: a) as taxas de acurácias dos modelos de classificação gerados com a base de dados ULA foram, em média, 66.46% inferiores para as características derivadas do FFT e 58.94% para as características derivadas do *Wavelet*. b) as taxas de acurácias dos modelos de classificação sobre a base de dados UA reduziram em média 22.43%, quando comparado aos resultados obtidos pelos classificadores usando somente as características do domínio do tempo.



**Figura C-2. Acurácia dos modelos de classificação baseado nas características de frequência derivadas do FFT. Fonte: autor.**



**Figura C-3. Acurácia dos modelos de classificação baseados nas características de frequência derivadas do Wavelet. Fonte: autor.**

Além disso, é possível destacar que as características derivadas do *Wavelet* são melhores que as características derivadas do FFT, confirmando os experimentos realizados por Figo et al. [37].

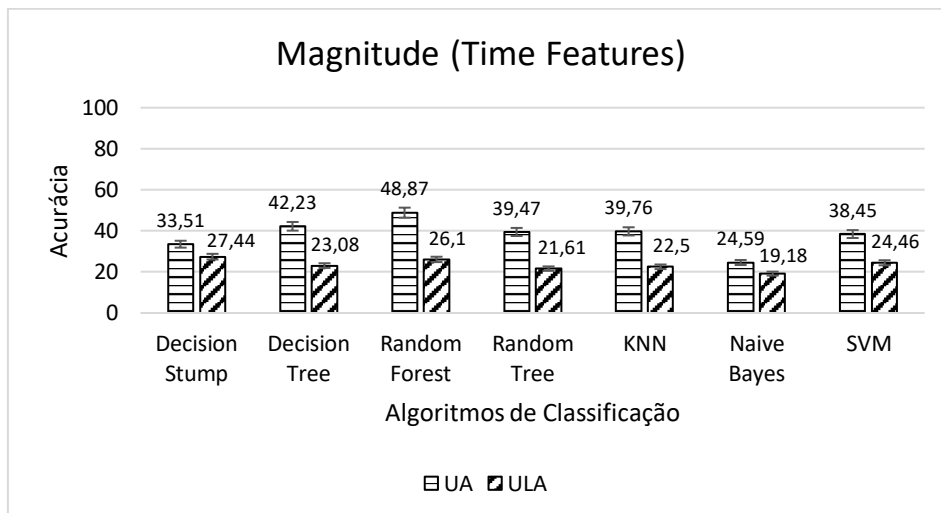
#### *Características Independentes de Orientação*

As características independentes de orientação são formadas pelo encadeamento das características de baixo nível, magnitude e componentes verticais e horizontais, com as características do domínio do tempo.

A Figura C-4 mostra as taxas de acurácias dos modelos de classificação gerados com as características do domínio do tempo derivadas das características de baixo nível magnitude. Os resultados mostram que as taxas de acurácias foram, em média, 51.74% inferiores quando comparadas aos experimentos realizados com as características do domínio do tempo. Como esperado, os modelos de classificação gerados a partir da base de dados UA obtiveram melhores taxas de acurácias. Do mesmo modo, A Figura C-5 mostra os experimentos realizados com os modelos de classificação gerados com as características do domínio do tempo derivadas das

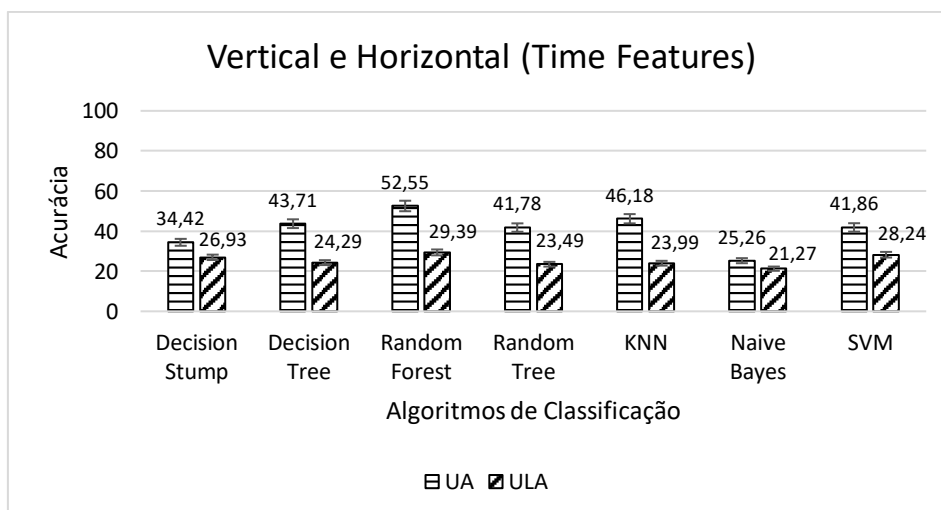


características componentes verticais e horizontais. Os resultados mostram uma acurácia média de 48.81% inferior aos modelos de classificação gerados com as características do domínio do tempo.



**Figura C-4.** Acurácia dos modelos de classificação baseados nas características de tempo derivadas da magnitude. Fonte: autor.

Os resultados descritos até agora foram executados com as características dos domínios do tempo e da frequência de forma separada. Com intuito de verificar as acurácias dos modelos de classificação gerados com a combinação desses dois domínios de características, foram realizados experimentos utilizando apenas o método de classificação *Random Forest*. Os resultados mostram que a união desses dois domínios em um único vetor de características apresentou uma taxa de acurácia de 90.10% em contraste com as acurácias de 90.63% e 78.87% obtidas pelos domínios do tempo e da frequência usados de forma separada. Isso mostra que somente as características do domínio do tempo são suficientes para representar as atividades físicas dos usuários.



**Figura C-5.** Acurácia dos modelos de classificação baseados nas características de tempo derivadas dos componentes verticais e horizontais. Fonte: autor.

Além disso, também foi avaliada a combinação entre as características do domínio do tempo e as características derivadas da magnitude. Tal combinação foi proposta por Shoaib et al. [91], sendo que, a base de dados de treino possui as seguintes coordenadas:

$(x_i, y_i, z_i, magnitude)$ . Neste caso, os resultados mostram uma acurácia de 89.8% utilizando o algoritmo *Random Forest*, ou seja, cerca de 1% inferior aos resultados obtidos somente com as características do domínio do tempo que foi de 90.63%.

A desvantagem das abordagens que combinam as características dos domínios do tempo/frequência e as características do domínio do tempo/magnitude é que, além de obterem menores acurácias, geram bases de dados com alta dimensão impactando no tempo de geração dos modelos de classificação.

### Discussões

Os modelos de classificação gerados com as características do domínio de tempo pelos métodos de aprendizagem de máquina, da família dos algoritmos das árvores de decisão, obtiveram as melhores taxas de acurácias utilizando a base de dados UA. Com destaque para o algoritmo *Random Forest* que obteve a melhor acurácia. Baseado nesses resultados, as seguintes conclusões podem ser apresentadas:

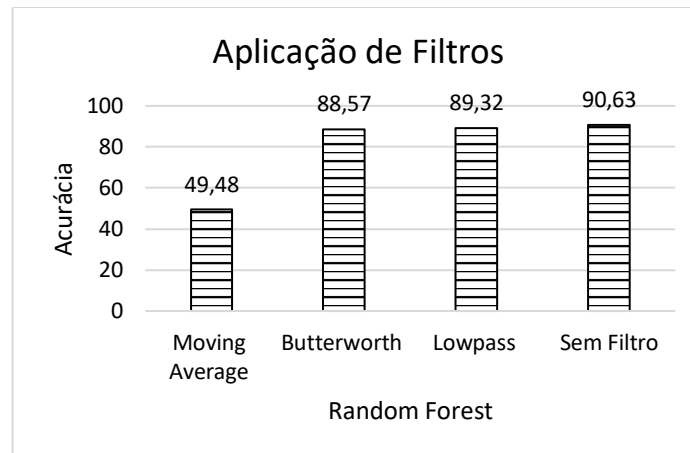
1. Os dados do acelerômetro linear, sem as informações de gravidade, não são adequados para reconhecer as atividades físicas dos usuários. Shoaib et al. [91] também confirmaram os mesmos resultados com o sensor localizado na cintura ou no bolso do usuário.
2. As características independentes de orientação não são adequadas para reconhecer as atividades físicas dos usuários quando utilizadas isoladamente.
3. Somente as características do domínio do tempo são suficientes para gerar bons modelos de classificação com os algoritmos de aprendizagem de máquina rasos. Além disso, essas características um menor custo em termos de processamento de dados [86].
4. As características derivadas do Wavelet são melhores que as características derivadas do FFT.
5. No contexto da análise de dados dos sensores inerciais, os algoritmos da família das árvores de decisão geram as melhores acurácias.

#### 1.2.2 - 2º Cenário: Técnicas de Redução de Ruídos dos Sinais

Esta seção apresenta resultados referentes aos estudos realizados com a aplicação de filtros sobre os sinais dos sensores inerciais na tentativa de eliminar ruídos e, conseqüentemente, melhorar a acurácia dos modelos de classificação. Nesta etapa, foram eliminados dos experimentos a base de dados ULA, as características independentes de orientação e as características do domínio de frequência. Além disso, os experimentos foram executados apenas com o melhor algoritmo de classificação, o *Random Forest*. Portanto, a configuração desse experimento é o seguinte:

- **Base de dados:** UCI-HAR UA com 1.311.439 amostras.
- **Tamanho da janela:** 5 segundos;
- **Sobreposição:** 0%;
- **Características:** Domínio do tempo;
- **Método de classificação:** *Random Forest*. Os parâmetros utilizados foram os padrões disponíveis na ferramenta Weka como 10 árvores com profundidade ilimitadas;
- **Filtros:** *Moving Average*, *Butterworth* e *Lowpass*. Tais filtros foram escolhidos devido a serem os mais utilizados na literatura de HAR [110] [29] [79].

A Figura C-6 apresenta as taxas de acurácia dos modelos de classificação gerados pelo algoritmo *Random Forest* sobre as características do domínio do tempo. Como pode ser observado, o filtro *Moving Average* obteve a pior taxa de acurácias sendo 49.38%. O motivo dessa taxa é devido à perda de informações causada pela redução do número de amostras resultantes das médias consecutivas de 3 pontos ao longo do sinal. Por outro lado, o filtro *Lowpass* obteve a melhor taxa de acurácia com 89.32%. Essa taxa é inferior à taxa obtida com as características do domínio do tempo sem a aplicação de filtros.



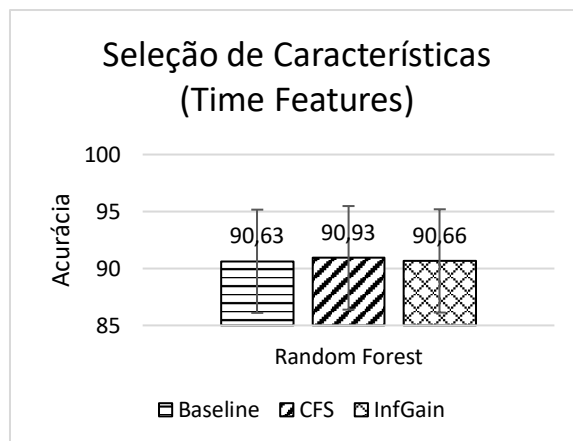
**Figura C-6. Resumo das taxas de acurácias dos modelos de classificação gerados pelo algoritmo Random Forest com e sem a aplicação de filtros. Fonte: autor.**

### 1.2.3 - 3º Cenário: Redução de Dimensionalidade

Esta seção apresenta resultados referentes aos estudos realizados com os métodos de redução de dimensionalidade com o propósito de selecionar as características mais representativas capazes de melhorar, ainda mais, a acurácia dos modelos de classificação. Nesse experimento, não foram incluídas as características independentes de orientação e as características do domínio de frequência, visto que os experimentos anteriores mostram que ambos os conjuntos de características resultam em taxas de acurácias menores que as características do domínio do tempo, quando usadas isoladamente. Portanto, a configuração desse experimento é o seguinte:

- **Base de dados:** UCI-HAR UA com 1.311.439 amostras.
- **Tamanho da janela:** 5 segundos;
- **Sobreposição:** 0%;
- **Características:** Domínio do tempo;
- **Método de classificação:** *Random Forest*. Os parâmetros utilizados foram os padrões disponíveis na ferramenta Weka como 10 árvores com profundidade ilimitadas;
- **Métodos de redução de dimensionalidade:** *Correlation-based Feature Selection* (CFS), ganho de informação (*Info-Gain*), PCA, LDA e KDA.

Os experimentos realizados neste cenário utilizam a taxa de acurácia do *Random Forest* (90.63%) como *baseline* para fins de comparação após a aplicação dos métodos de redução de dimensionalidade.



**Figura C-7.**Resumo das acurácias dos modelos de classificação após a seleção de características. Fonte: autor.

A Figura C-7 mostra um resumo comparativo das taxas de acurácias dos modelos de classificação após a aplicação dos métodos de seleção de característica CFS e *Info-Gain*. Os resultados mostram um aumento médio nas taxas de acurácia de 0.15% em relação ao *baseline*. Em resumo, os métodos obtiveram os seguintes percentuais de eliminação das características irrelevantes: O CFS reduziu o número das características em 14%, obtendo uma taxa de acurácia de 90.93%. O método *InfoGain* reduziu o número de características em 16%, obtendo uma taxa de acurácia de 90.66%. Nessa perspectiva, conclui-se que os métodos de seleção de características reduziram o número de características em 15% de forma que os modelos de classificação obtiveram ganhos na acurácia.

Por outro lado, os experimentos realizados por Khan et al. [54] mostram que os métodos de redução de dimensionalidade PCA, LDA e KDA possuem melhor desempenho quando comparados com os métodos CFS e *InfoGain*. Dentre eles, o KDA combinado com o Kernel RBF (*Gaussian*) gera melhores resultados. Nesse trabalho, os experimentos foram realizados em uma base de dados diferente (coletada pelo autor) e a melhor acurácia alcançada foi de 94% comparada a 73% sem a aplicação de nenhum algoritmo de redução de dimensionalidade. Os modelos de classificação foram avaliados pela técnica *leave-one-subject-out*, o que explica a grande diferença entre as acurácias. Neste contexto, o uso do método KDA é vantajoso porque houve melhores significativas dos resultados no modo *leave-one-subject-out*.

#### 1.2.4 - 4º Cenário: Janela de Tempo, Sobreposição e Frequência

Esta seção apresenta resultados referentes aos estudos realizados com variações no tamanho das janelas de tempo, na taxa de sobreposição de janelas e na taxa de frequência da coleta dos dados. Neste cenário, foram realizados experimentos apenas com as variações no tamanho das janelas de tempo e na taxa de sobreposição. Pois a base de dados UCI-HAR não permite a variação da taxa de frequência devido os dados terem sido coletados somente a frequência de 50Hz. Por este motivo, os resultados referentes a variação da frequência são obtidos dos experimentos conduzidos por Khusainov et al. [23]. Portanto, a configuração utilizada na avaliação das janelas de tempo é a seguinte:

- **Método de classificação:** *Random Forest*. Os parâmetros utilizados foram os padrões disponíveis na ferramenta Weka como 10 árvores com profundidade ilimitadas;
- **Características:** Domínio do tempo;
- **Janelas de tempo:** 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 e 20 segundos;

- **Sobreposição:** 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% e 90%;

A Figura C-8 mostra o comportamento da acurácia dos modelos de classificação com as variações nas taxas de sobreposição para a janela de tempo de 5 segundos. A taxa de sobreposição de 50% gerou o modelo de classificação com maior taxa de acurácia de 92.33%. Além disso, o gráfico mostra que existe uma tendência de aumento na taxa de acurácia até 50% de sobreposição, após isso, ocorre uma diminuição no desempenho do classificador.

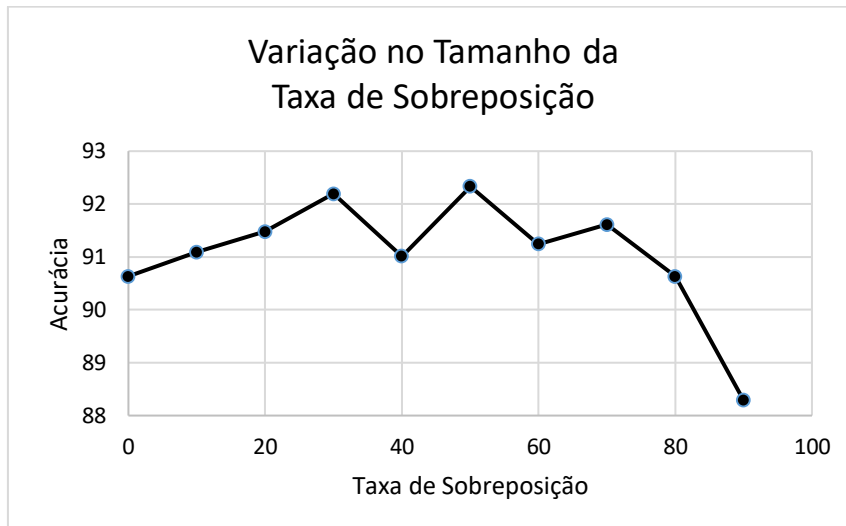


Figura C-8. Resultados das acurácias dos modelos de classificação com a variação na taxa de sobreposição. Fonte: autor.

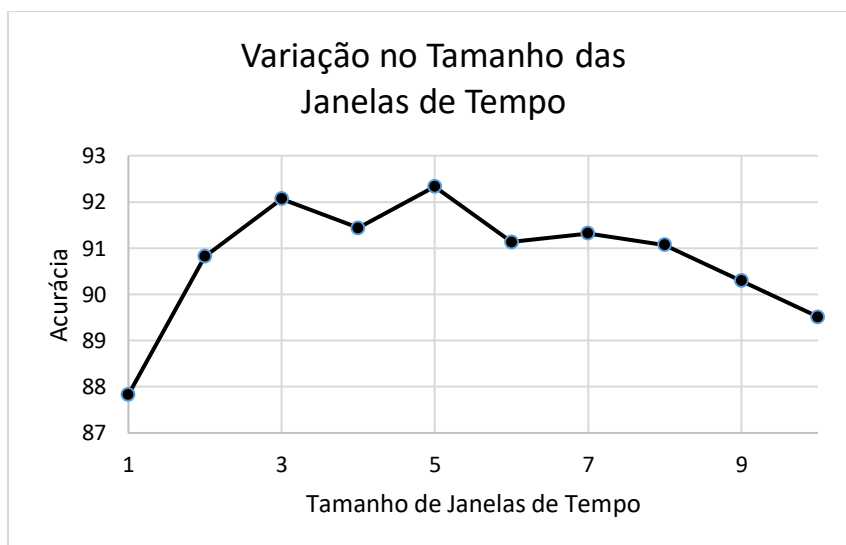


Figura C-9. Resultados das acurácias dos modelos de classificação com a variação da janela de tempo. Fonte: autor.

A Figura C-9 mostra o comportamento da acurácia dos modelos de classificação com as variações no tamanho da janela de tempo. Neste experimento, nós utilizamos a taxa de sobreposição de 50%. Como pode ser observado, a janela de tempo com tamanho de 5 segundos obteve a melhor acurácia, 92.33%. Além disso, o gráfico mostra que existe uma tendência de aumento da acurácia até o limite do tempo de 5 segundos, depois disso, a acurácia tende a cair. É notório destacar que a base de dados UA foi coletada a uma frequência de 50Hz e, portanto, uma janela de tempo com 5 segundos contém 250 amostras. Na literatura, os resultados

mostram que o tamanho ideal para as janelas de tempo fixas pode variar em torno de 2 a 5 segundos considerando uma frequência de 20Hz a 50 Hz [87] [92] [27] [23].

Com relação a frequência da coleta de dados, os experimentos realizados por Khusainov et al. [23] mostram que 20 Hz é suficiente para capturar informações úteis das atividades físicas dos usuários por meio dos sensores inerciais. As frequências testadas variam de 10 a 60 Hz numa base de dados própria do autor. Os trabalhos da literatura utilizam uma variação de frequência entre 1 Hz a 200 Hz, conforme trabalhos apresentados na Seção 2.6. É importante destacar que quanto maior for a frequência, maior será o consumo de energia do smartphone.

### Discussões

Os experimentos executados até agora utilizaram todas as características do domínio do tempo apresentadas na Seção 3.2.1. No entanto, a maioria dos trabalhos da literatura utiliza somente as características estatísticas simples como média, variância, desvio padrão, *Root Mean Square*, mínimo, máximo, amplitude e correlação. Nesse sentido, foram realizados experimentos apenas com essas características observando os melhores cenários identificados até agora, que incluem o algoritmo de classificação *Random Forest*, janela de tempo de 5 segundos e taxa de sobreposição de 50%. Para essa configuração, os modelos de classificação obtiveram uma acurácia foi de 92.48%, a maior acurácia alcançada até agora usando a estratégia *cross-validation*. Isso mostra que as 8 características simples citadas acima são suficientes para discriminar as atividades físicas dos usuários. Além disso, em termos de redução de dimensionalidade, tais características representam apenas 20% do total de características do domínio do tempo utilizadas nos nossos experimentos até o presente momento.

Portanto, uma vez que as configurações para gerar os melhores modelos de classificação são conhecidas, nós apresentamos mais detalhes referentes aos erros e acertos desses modelos separados por atividade. A Tabela C.2 mostra a matriz de confusão para a estratégia de avaliação *cross-validation* utilizando modelos de classificação gerados com as características média, variância, desvio padrão, *Root Mean Square*, mínimo, máximo, amplitude e correlação.

**Tabela C.2 Matriz de confusão para a estratégia Cross-validation.**

STANDING	SITTING	LAYING	WALKING	DOWNSTAIRS	UPSTAIRS	
1385	52	1	14	3	29	STANDING
79	1142	6	7	2	2	SITTING
18	10	1411	2	0	0	LAYING
2	0	6	1231	39	54	WALKING
1	0	0	42	779	53	DOWNSTAIRS
1	0	0	76	63	967	UPSTAIRS

Como pode ser observado na Tabela C.2, tanto os erros das atividades sem movimento (STANDING, SITTING e LAYING) quanto os erros das atividades com movimento (WALKING, DOWNSTAIRS, UPSTAIRS) são causados pela similaridade entre as atividades de cada grupo. Por exemplo, 79 amostras da atividade STANDING foram confundidas com a atividade SITTING e, 76 e 42 amostras da atividade WALKING foram confundidas com as atividades DOWNSTAIRS e UPSTAIRS, respectivamente. Esses erros ocorrem devido os dados entre tais atividades serem muito parecidos uns com os outros. Além disso, a atividade que se mostra mais estável, sem muitos erros, é a atividade LAYING.

### 1.2.5 - 5º Cenário: Outras Bases de Dados

Esta seção mostra exemplos de acurácias dos melhores modelos de classificação em outras bases de dados públicas como WISDM e SHOAIB descritas no Apêndice A. Tais modelos foram gerados com as seguintes configurações:

- **Bases de dados:** UCI-HAR UA (1.311.439 amostras), WISDM (1.098.830 amostras) e SHOAIB (629.977 amostras). Foram utilizados os dados somente do acelerômetro e o smartphone localizado na cintura do usuário;
- **Tamanho da janela:** 5 segundos;
- **Sobreposição:** 50%;
- **Características:** média, variância, desvio padrão, *Root Mean Square*, mínimo, máximo, amplitude e correlação;
- **Métodos de classificação:** *Random Forest*. Os parâmetros utilizados foram os padrões disponíveis na ferramenta Weka como 10 árvores com profundidade ilimitadas;

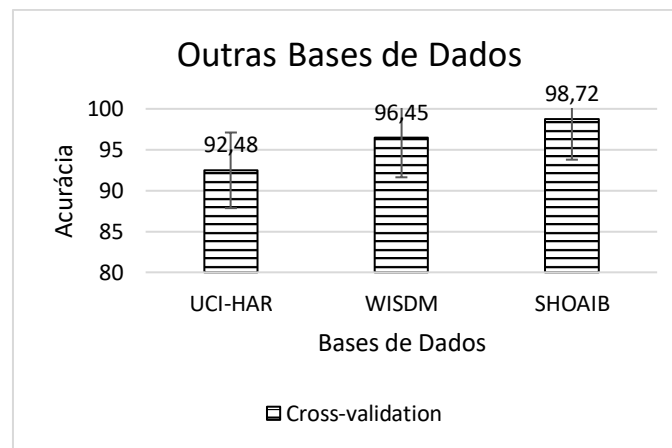


Figura C-10. Exemplos de acurácia dos modelos de classificação utilizando diferentes bases de dados. Fonte: autor.

A Figura C-10 mostra o resumo das taxas de acurácias dos modelos de classificação gerados a partir das três bases de dados utilizadas. Como pode ser observado, as bases de dados WISDM e SHOAIB obtiveram melhores modelos de classificação comparado com a base de dados UCI-HAR UA. Isso ocorreu devido as bases de dados WISDM e SHOAIB possuírem dados mais homogêneos, ou seja, sem muitos ruídos que podem interferir no desempenho dos modelos de classificação.

## 1.3. Fusão de Dados

Esta seção apresenta os resultados referentes aos estudos realizados por meio da fusão de dados entre os sensores inerciais acelerômetro e giroscópio. Neste cenário, a estratégia de fusão de dados utilizada é empregada após a etapa de extração de características por meio da técnica de concatenação dos vetores de características explicada na Seção 3.7. Neste experimento, foi utilizada outra versão da base de dados UCI-HAR que contém informações de dois sensores, acelerômetro e giroscópio. Essa nova versão está descrita no Apêndice A e é conhecida como a base de dados HAPT.

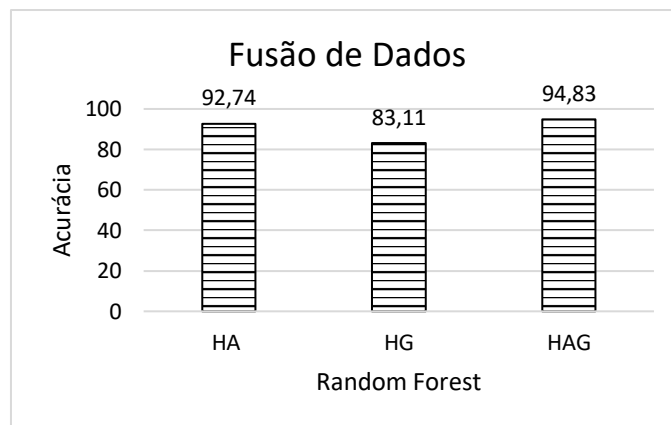
Os experimentos foram executados sob três condições. A primeira condição inclui o uso completo da base de dados HAPT com as informações dos dois sensores (HAG). A segunda condição inclui o uso parcial da base de dados HAPT com os dados somente do sensor acelerômetro (HA). A terceira condição inclui o uso parcial da base de dados HAPT com os dados somente do sensor giroscópio (HG). O objetivo destes experimentos é avaliar o impacto da acurácia dos modelos de classificação por meio da análise do uso isolado dos dados de cada sensor e da combinação entre eles. Além disso, os experimentos possuem as seguintes configurações:

- **Método de classificação:** *Random Forest*. Os parâmetros utilizados foram os padrões disponíveis na ferramenta Weka como 10 árvores com profundidade ilimitadas;
- **Características:** média, variância, desvio padrão, *Root Mean Square*, mínimo, máximo, amplitude e correlação;
- **Janela de tempo:** 5 segundos;
- **Sobreposição:** 50%;

A Figura C-11 mostra uma queda em torno de 11% nas taxas de acurácia dos modelos de classificação que utilizam somente os dados do giroscópio (HG) comparado aos modelos gerados com os dados do acelerômetro (HA). Por outro lado, a combinação dos dados desses dois sensores gera modelos de classificação mais precisos com um aumento médio de 2% na taxa de acurácia alcançando 94.83%. Wang et al. [93] obtiveram resultados semelhantes com a análise da combinação dos dados dos sensores acelerômetro e giroscópio. Os resultados mostram que o acelerômetro sempre predomina em relação ao giroscópio, em todos os casos, mesmo quando ambos são utilizados isoladamente. Em relação à fusão de dados, as acurácias dos modelos de classificação também subiram em média 2%. Além disso, os resultados mostram que a fusão entre esses dois sensores é mais precisa para o reconhecimento de atividades estáticas (e.g. em pé e sentado) quando comparada com as atividades dinâmicas (e.g. andar e correr).

Shoib et al. [91] também analisaram a combinação dos dados dos sensores acelerômetro, giroscópio e magnetômetro. Os resultados mostram que a combinação dos dados do acelerômetro e o giroscópio melhoram a acurácia dos modelos de classificação em 2%. Os experimentos concluíram que o acelerômetro é o sensor predominante para as atividades de movimento, exceto para as atividades subir e descer escadas onde o giroscópio apresentou melhor desempenho. O magnetômetro, por sua vez, foi avaliado de forma isolada e, em geral, os dados geram modelos com baixas taxas de acurácia. Porém, a correta seleção das características pode melhorar os resultados significativamente.





**Figura C-11. Comparação das acurácias dos modelos de classificação com a combinação dos dados do acelerômetro e giroscópio. Fonte: autor.**

Dessa forma, conclui-se que o uso de um único sensor limita o escopo do problema de HAR e que, conforme aumenta o número de atividades, mais fontes de dados devem ser utilizadas nas soluções de HAR. Caso contrário, as acurácias tendem a cair.

#### 1.4. Posição do Smartphone no Corpo do Usuário

A coleta de dados da UCI-HAR e HAPT foi realizada somente com o smartphone localizado na posição da cintura dos usuários. Por esse motivo, essa base de dados não permite realizar experimentos no contexto das posições dos smartphones do corpo do usuário. Para preencher essa lacuna, nós apresentamos alguns trabalhos relevantes encontrados na literatura que realizaram experimentos detalhados com as seguintes posições: cintura, braço, mão, pulso e bolsos da calça.

Chen e Shen [28] analisaram o impacto das posições do smartphone no contexto das atividades físicas. Os experimentos avaliaram, de forma separada, todas as posições do smartphone no corpo do usuário. Os resultados mostram que, em geral, o bolso da calça é a melhor posição e a mão é a pior posição para reconhecer as atividades simples de movimento. Porém, o autor destaca que as posições devem ser escolhidas dependendo do contexto das atividades. Por exemplo, a posição mais adequada para reconhecer as atividades nadar e abrir porta é no pulso ou na mão do usuário. O terreno também influencia nos resultados, por exemplo, existe diferença entre o usuário correr numa trilha de uma floresta e o mesmo usuário correr na calçada de uma praça.

Shoaib et al. [91] analisaram a eficiência dos sensores acelerômetro e giroscópio em diversas posições do smartphone no corpo do usuário. Em geral, o sensor acelerômetro gera melhores resultados com o smartphone localizado na cintura e nos bolsos dos usuários e, também, quando os dados de diferentes posições são combinados. Um caso particular identificado foi que o sensor giroscópio reconhece melhor as atividades de subir e descer escada com o sensor localizado, principalmente, no braço e nos bolsos dos usuários. O autor conclui que a combinação dos dados dos sensores e das posições não traz melhorias nas acurácias dos modelos de classificação. Outro trabalho realizado por Shoaib et al. [27] analisa o impacto da localização dos smartphones, no pulso e do bolso da calça do usuário, na acurácia dos modelos de classificação no contexto das atividades de movimento. Além das atividades simples de movimento como andar e correr, este trabalho reconhece outras atividades mais complexas como tomar café e escrever. De acordo com os autores, existe uma maior probabilidade dos

smartphones, localizados no pulso, capturar os movimentos dessas atividades com mais eficiência. Neste contexto, os resultados mostram que a combinação dos dados dos smartphones localizados nessas duas posições melhora a acurácia dos modelos de classificação.

## 1.5. Aprendizagem Profunda

Esta seção apresenta os resultados referentes aos estudos realizados para fins de comparação entre os métodos de aprendizagem de máquina rasos e profundos. Os resultados foram extraídos dos estudos mapeados por Wang et al. [24]. Os experimentos selecionados foram realizados com a mesma base de dados utilizada em nossos experimentos, a UCI-HAR. Além disso, a estratégia de avaliação dos modelos utilizada foi a *cross-validation*.

A Figura C-12 mostra um resumo comparativo entre a melhor acurácia alcançada pelos modelos de classificação *Random Forest*, com janela de tempo de 5 segundos e sobreposição de 50%, e as melhores acurácias dos modelos de classificação baseados na abordagem de aprendizagem profunda identificadas na literatura. Como pode ser observado, as acurácias dos modelos de aprendizagem profunda são maiores, em média, 2.5% comparadas à acurácia do melhor modelo de classificação gerados pelos algoritmos de aprendizagem de máquina rasos. Baseado nesses resultados, é possível afirmar que o melhor modelo de classificação de atividades físicas, no contexto dos smartphones com sensores inerciais, é baseado na arquitetura da rede neural *Stacked Autoencoder (SAE)* com uma acurácia de 97.5%. Este modelo foi proposto por Almaslukh et al. [94] e comparado com o modelo estado da arte dos métodos rasos gerado pelo algoritmo *One-Vs-One Multiclass Linear SVM* com uma acurácia de 96.4% [224]. Portanto, os algoritmos de aprendizagem profunda estão sendo considerados, atualmente, o estado da arte das soluções de HAR baseada em smartphones com sensores inerciais.

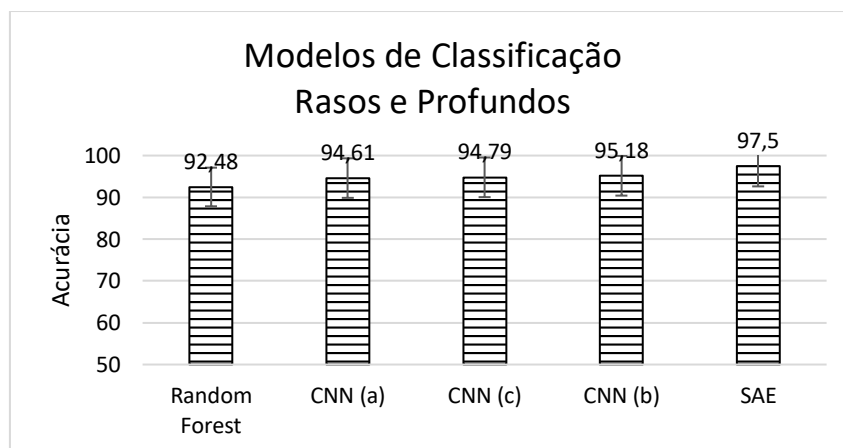


Figura C-12. Comparação entre as taxas de acurácia dos modelos de classificação dos algoritmos de aprendizagem rasa e profunda. Fonte: autor.

# APÊNDICE D. FERRAMENTA BHAR-BENCHMARKING HAR

A BHAR consiste em uma ferramenta desenvolvida durante esta tese de doutorado que contém implementado a maioria dos *baselines* identificados na literatura para reconhecer as atividades humanas no contexto da análise de dados de sensores inerciais dos smartphones. A ferramenta BHAR foi criada com intuito de facilitar a comparação de diferentes abordagens da literatura. Além disso, as implementações referentes a cada abordagem podem ser facilmente reaproveitadas para gerar aplicações específicas capazes de executar nos smartphones. A Figura D-1 mostra a tela de trabalho principal da ferramenta.



**Figura D-1. Tela principal da ferramenta BHAR.**

A ferramenta BHAR possui os seguintes módulos para manipulação de dados:

1. Pré-processamento da base de dados incluindo manipulação de variáveis, conversão e junção de arquivos.
2. A visualização dos dados é mostrada em gráficos ou blocos de processamento. A visualização de gráficos é útil para verificar o comportamento dos dados antes e após o processamento. A visualização em blocos é útil para manipular bases de dados grandes que requerem bastante uso da memória RAM.
3. Técnicas de filtro de sinais, como Média Móvel, Lowpass e Butterworth citadas no Capítulo 3.
4. Técnicas de fusão de dados, como magnitude e PCA.
5. Técnicas de extração de características no domínio do tempo, frequência e discreto descritas na Seção 3.4.

6. Treinamento de modelos de classificação por meio dos algoritmos de aprendizagem de máquina e algoritmos de classificação simbólicos descritos no Capítulo 4. A Figura D-2 mostra uma visão geral da tela de treinamento e classificação dos algoritmos.
7. Módulo que implementa as etapas de processamento de dados, visualização de dados e treino e teste do algoritmo NOHAR. A Figura D-3 mostra uma visão geral da tela que implementa o NOHAR.

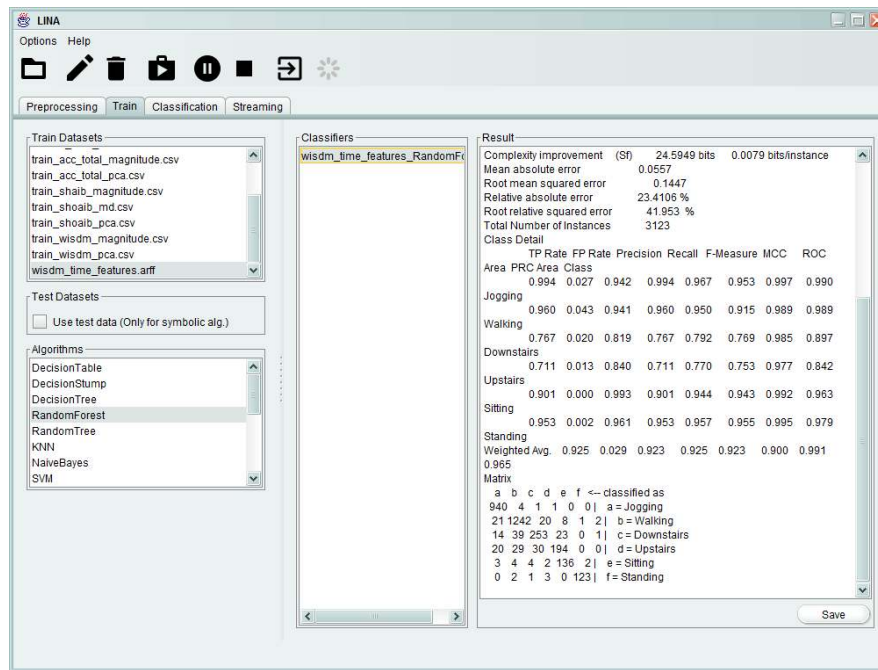


Figura D-2. Tela de Treinamento e classificação.

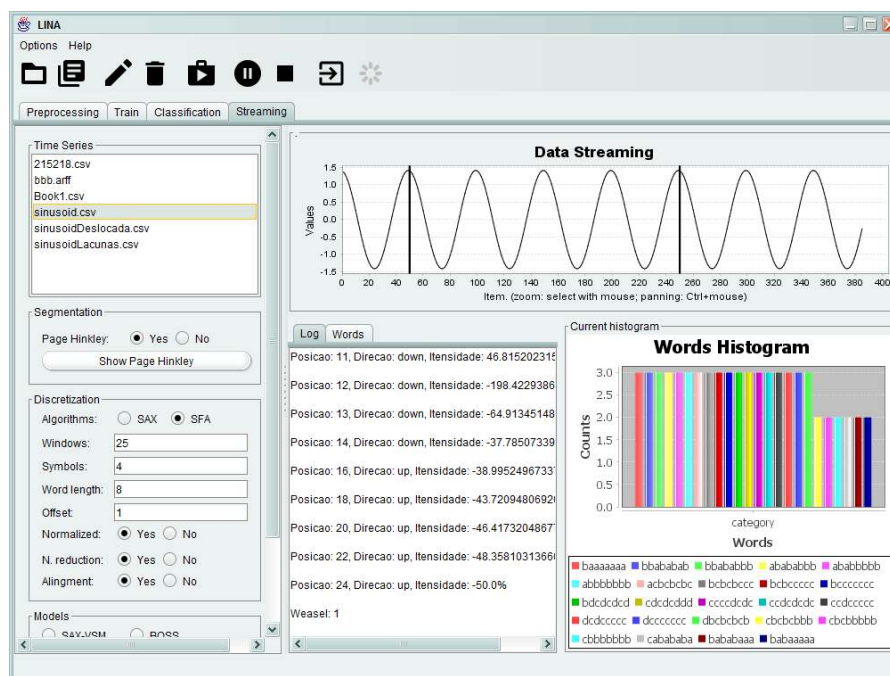


Figura D-3. Tela do algoritmo NOHAR.