UFAM

UNIVERSIDADE FEDERAL DO AMAZONAS

FACULDADE DE TECNOLOGIA - FT

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA -PPGEE

EXPLICAÇÕES VISUAIS APLICADAS A REDES NEURAIS CONVOLUCIONAIS UNIDIMENSIONAIS COM ÊNFASE NO RECONHECIMENTO HUMANO DE ATIVIDADES

Manaus – AM

Junho de 2024

UFAM

GUSTAVO DE AQUINO E AQUINO

EXPLICAÇÕES VISUAIS APLICADAS A REDES NEURAIS CONVOLUCIONAIS UNIDIMENSIONAIS COM ÊNFASE NO RECONHECIMENTO HUMANO DE ATIVIDADES

Tese apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Faculdade de Tecnologia da Universidade Federal do Amazonas, Campus Universitário Senador Arthur Virgílio Filho, como requisito parcial para a obtenção do título de Doutor em Engenharia Elétrica.

ORIENTADORA: PROFA DRA. MARLY GUIMARÃES FERNANDES COSTA

COORIENTADOR: PROF DR. CÍCERO FERREIRA FERNANDES COSTA FILHO

Manaus – AM

Junho de 2024

Ficha Catalográfica

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

Aquino, Gustavo de Aquino e
A657e Explicações visuais aplicadas a redes neurais convolucionais unidimensionais com ênfase no reconhecimento humano de atividades / Gustavo de Aquino e Aquino . 2024
148 f.: 31 cm.

Orientadora: Marly Guimarães Fernances Costa
Coorientador: Cícero Ferreira Fernandes Costa Filho
Tese (Doutorado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Reconhecimento humano de atividades. 2. Aprendizado profundo. 3. Redes neurais convolucionais unidimensionais. 4. Inteligência artificial explicável. 5. Dados de acelerômetro. I. Costa, Marly Guimarães Fernances. II. Universidade Federal do Amazonas III. Título

Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica

FOLHA DE APROVAÇÃO

Poder Executivo Ministério da Educação
Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica

Pós-Graduação em Engenharia Elétrica. Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Campus Universitário, Setor Norte - Coroado, Pavilhão do CETELI. Fone/Fax (92) 99271-8954 Ramal:2607. E-mail: ppgee@ufam.edu.br

GUSTAVO DE AQUINO E AQUINO

EXPLICAÇÕES VISUAIS APLICADAS A REDES NEURAIS CONVOLUCIONAIS UNIDIMENSIONAIS COM ÊNFASE NO RECONHECIMENTO HUMANO DE ATIVIDADES

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovada em 07 de junho de 2024.

BANCA EXAMINADORA
Profª. Dra. Marly Guimarães Fernandes Costa- Presidente
Prof. Dr. Vicente Ferreira de Lucena Junior - Membro Titular 4 - Interno
Prof. Dr. Eduardo James Pereira Souto - Membro Titular 3 - Externo
Profª. Dra Fabíola Guerra Nakamura - Membro Titular 2 - Externo
Profª. Dra Letícia Rittner - Membro Titular 1 - Externo
Documento assinado eletronicamente

Manaus, 28 de maio de 2024.

Documento assinado eletronicamente por **Marly Guimarães Fernandes Costa**, **Professor do Magistério Superior**, em 12/06/2024, às 08:14, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Vicente Ferreira de Lucena Júnior**, **Professor do Magistério Superior**, em 12/06/2024, às 10:48, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Fabíola Guerra Nakamura**, **Professor do Magistério Superior**, em 17/06/2024, às 09:59, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Eduardo James Pereira Souto**, **Professor do Magistério Superior**, em 17/06/2024, às 10:30, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Leticia Rittner**, **Usuário Externo**, em 20/06/2024, às 09:10, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2072278** e o código CRC **346FD07E**.

Av. Octávio Hamilton Botelho Mourão - Bairro Coroado 1 Campus Universitário Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) 3305-1181
CEP 69080-900 Manaus/AM - mestrado_engeletrica@ufam.edu.br

Referência: Processo nº 23105.020803/2024-93 SEI nº 2072278

Criado por 31183646291, versão 4 por 31183646291 em 28/05/2024 14:02:44.

À minha amada esposa, Kimberly Mouzinho, que sempre esteve ao meu lado, fornecendo amor, apoio e compreensão durante toda a jornada da minha tese.

Agradeço aos meus pais, Lazaro Barros e Hermina Aquino, por acreditarem em mim e me inspirarem a seguir meus sonhos e objetivos acadêmicos.

Um agradecimento especial à Profa. Dra. Marly Guimarães e Prof. Dr. Cícero Filho, meus orientadores, por compartilharem seu conhecimento, orientação e paciência, ajudando-me a moldar e aprimorar meu trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES).

Esta pesquisa, realizada no âmbito do Projeto Samsung-UFAM de Ensino e Pesquisa (SUPER), de acordo com o Artigo 39 do Decreto n°10.521/2020, foi financiada pela Samsung Eletrônica da Amazônia Ltda, nos termos da Lei Federal n°8.387/1991, através do convênio 001/2020 firmado com a UFAM e FAEPI, Brasil.

Por fim, gostaria de agradecer aos meus amigos de trabalho, que tornaram o processo mais agradável e enriquecedor, proporcionando momentos de camaradagem, discussões e aprendizado mútuo.

A todos vocês, meu mais sincero agradecimento.

"O verdadeiro sinal de inteligência não é o conhecimento, mas a imaginação."
(Albert Einstein)

Resumo

A Reconhecimento Humano de Atividades (HAR) é uma área de crescente importância, especialmente com a popularização dos dispositivos vestíveis. Redes neurais convolucionais unidimensionais (1D CNNs) surgem como uma abordagem amplamente utilizada para HAR. Essas arquiteturas são impulsionadas por dados e têm a capacidade de aprender padrões complexos nos sinais que podem ser usados para classificar atividades humanas. No entanto, embora 1D CNNs possam alcançar resultados numéricos expressivos, compreender e explicar a tomada de decisão desses modelos ainda é um desafio. Este trabalho propõe duas abordagens inovadoras para gerar explicações visuais em 1D CNNs aplicadas ao HAR, a primeira utilizando *Gradient-weighted Class Activation Mapping* (grad-CAM) e a segunda com *t-Distributed Stochastic Neighbor Embedding* (t-SNE). Nosso objetivo é, por meio das visualizações, entender e explicar os padrões complexos aprendidos pelos modelos 1D CNNs durante o processo de treinamento, propondo visualizações que possibilitam o entendimento do processo de tomada de decisão dos modelos. As explicações propostas permitem a identificação de vieses nos modelos e nos conjuntos de dados, a análise de como a abordagem de validação impacta o aprendizado do modelo e também a escolha de um melhor modelo, observando não apenas resultados numéricos, mas também qualitativos. Em geral, com base nos experimentos propostos neste trabalho, a combinação de técnicas de inteligência artificial explicável com aprendizado profundo tem potencial para proporcionar uma visão holística sobre a capacidade dos modelos treinados, tornando possível criar explicações e formular hipóteses sobre a tomada de decisão dos modelos.

Palavras-chave: Reconhecimento humano de atividades; Aprendizado profundo; Redes neurais convolucionais unidimensionais; Inteligência artificial explicável; Dados de acelerômetro; grad-CAM; t-SNE.

Abstract

HAR is an area of growing importance, especially with the popularization of wearable devices. One-dimensional convolutional neural networks (1D CNNs) have emerged as a widely used approach for HAR. These architectures are data-driven and have the ability to learn complex patterns in signals that can be used to classify human activities. However, although 1D CNNs can achieve impressive numerical results, understanding and explaining the decision-making of these models remains a challenge. This work proposes two innovative approaches to generate visual explanations in 1D CNNs applied to HAR, the first one utilising grad-CAM and the second one utilizing t-SNE. Our goal is, through visualizations, to understand and explain the complex patterns learned by 1D CNN models during the training process, proposing visualizations that enable the understanding of the models' decision-making process. The proposed explanations allow the identification of biases in the models and datasets, analysis of how the validation approach impacts model learning, and also the selection of a better model by observing not only numerical but also qualitative results. In general, based on the experiments proposed in this work, the combination of explainable artificial intelligence techniques with deep learning has the potential to provide a holistic view of the capabilities of trained models, making it possible to create explanations and formulate hypotheses about the models' decision-making process.

Keywords: Human Activity Recognition; Deep Learning; One-Dimensional Convolutional Neural Networks; Explainable Artificial Intelligence; Accelerometer Data; grad-CAM; t-SNE.

Lista de Figuras

Lista de Tabelas

Acrónimos e nomenclaturas

1D CNN Rede Neural Convolucional Unidimensional

ADL Actividades Diárias

ANN Rede Neural Artificial

BUI Identificação Biométrica do Usuário

CNN Rede Neural convolucional

DL Aprendizado Profundo

FN *Falso Negativo*

FP *Falso Positivo*

GPS Sistema de Posicionamento Global

grad-CAM *Gradient-weighted Class Activation Mapping*

HAR Reconhecimento Humano de Atividades

IA Inteligência Artificial

IMU Unidade de Medição Inercial

KL Kullback-Leibler

LIME Local Interpretable Model-agnostic Explanations

LOSO *Leave-one-subject-out*

LSTM Long short-term memory

ML Aprendizado de Máquina

PCA Análise de Componentes Principais

PT Transições Posturais

ReLU Função de Ativação Linear Retificada

SD Sujeito Dependente

SHAP SHapley Additive exPlanations

SI Sujeito Independente

t-SNE *t-Distributed Stochastic Neighbor Embedding*

TAF Taxa de aceitação falsa

TN *Verdadeiro Negativo*

TP *Verdadeiro Positivo*

TPR Taxa de Verdadeiro Positivo

TV Taxa de Verificação

XAI Inteligência Artificial Explicável

YOLO *You Only Look Once*

Sumário

1

Introdução

Smartphones e dispositivos vestíveis oferecem oportunidades sem precedentes para monitorar sinais fisiológicos humanos, criando possíveis aplicações em áreas como Reconhecimento Humano de Atividades (HAR), vida assistida por ambiente, saúde e outras (Booth *et al.*, 2012, Zhang *et al.*, 2022). O mercado de tecnologia vestível está em alta, com previsão de atingir US$ 415,12 bilhões até 2029 (Research, 2022). Esse crescimento é impulsionado por fatores como tecnologias inovadoras, aumento da prevalência de doenças crônicas e obesidade, além da demanda por dispositivos compactos que integram todas as funções de computação (Research, 2022). Nas últimas décadas, diversos artigos foram publicados, propondo novas técnicas e soluções em HAR, usando a ampla variedade de sensores incorporados em um *smartphone*, como acelerômetros, giroscópios, microfones, unidades de Sistema de Posicionamento Global (GPS) e outros (Ferrari *et al.*, 2021, Hayat *et al.*, 2022, Zhang *et al.*, 2022). Dispositivos móveis usam esses sensores para capturar dados e desenvolver assistentes pessoais para monitorar e ajudar os usuários em suas rotinas de exercícios e atividades, tornando os usuários mais cientes sobre o seu condicionamento físico ou nível de sedentarismo (Bragança *et al.*, 2022, Zhang *et al.*, 2022).

A Inteligência Artificial (IA) forma a base de qualquer tecnologia usada para o HAR (Gupta *et al.*, 2022, Hayat *et al.*, 2022). Sistemas HAR identificam automaticamente atividades humanas usando dados capturados por dispositivos vestíveis ou *smartphones* (Ferrari *et al.*, 2021, Zhang *et al.*, 2022). HAR tem inúmeras aplicações no setor de saúde e bem-estar para indivíduos que desejam monitorar sua saúde e condicionamento físico (Ferrari *et al.*, 2021, Hayat *et al.*, 2022, Kazemimoghadam & Fey, 2022, Zhang *et al.*, 2022).

Sensores de Unidade de Medição Inercials (IMUs) se destacam na área de HAR. IMUs são conjuntos eletrônicos capazes de reportar variações de posição, fornecendo

suas detecções como séries temporais. Geralmente, IMUs são compostos por um pacote de sensores integrados que inclui acelerômetro, giroscópio e magnetômetro. O acelerômetro se destaca como o sensor mais importante para desenvolver algoritmos de HAR. O sensor pode detectar movimentos de alta e baixa frequência (Ferrari *et al.*, 2021, Gu *et al.*, 2021). Um giroscópio e um magnetômetro podem ser usados para complementar os dados do acelerômetro (Ferrari *et al.*, 2021, Zhang *et al.*, 2022). Quanto mais informações forem passadas a um sistema de IA, mais fácil será aprender um padrão. No entanto, em aplicações práticas, mais sensores significam mais recursos computacionais, que geralmente podem não estar disponíveis em um dispositivo com capacidades limitadas, como um *smartwatch* ou *smartphone*. Desenvolver uma solução que utiliza apenas um tipo de sensor torna-a mais apropriada para uso em cenários do mundo real.

Aprendizado Profundo (DL) é um campo do Aprendizado de Máquina (ML). Recentemente, o DL superou o ML em muitas aplicações, como o caso de classificação baseado em dados de série temporal (Zhang *et al.*, 2022). ML tradicional requer um especialista para extrair informações de alta relevância a partir dos sensores. No entanto, isso exige uma quantidade significativa de esforço manual (Ferrari *et al.*, 2021, Zhang *et al.*, 2022). Em contraste, temos modelos de DL que podem extrair características. As camadas ocultas de um modelo profundo permitem a ele extrair características de alto nível dos dados, tão boas quanto, se não melhores do que modelos clássicos (Ferrari *et al.*, 2021), dando ao DL uma vantagem sobre ML tradicional. Embora seja conhecida a capacidade dos modelos profundos, em HAR há uma grande quantidade de estudos que utilizam ML tradicional, não explorando o total poder dos sensores de série temporal (Ferrari *et al.*, 2021, Zhang *et al.*, 2022).

As Redes Neurais Convolucionais (CNNs) estão entre os modelos de DL mais utilizados e surgiram como o método de ponta para inúmeras tarefas. Nos últimos anos, Redes Convolucionais Unidimensionais (1D CNNs) mostraram-se eficazes na classificação de atividade humana a partir de sensores vestíveis, como acelerômetros e giroscópios (Chen *et al.*, 2021, Ferrari *et al.*, 2021, Gu *et al.*, 2021, Zhang *et al.*, 2022). As 1D CNNs são semelhantes às redes neurais convolucionais padrão, mas, em vez de processar imagens bidimensionais, processam sinais unidimensionais, como dados de séries temporais. As 1D CNNs podem ser treinadas de ponta a ponta para extrair características diretamente dos dados brutos.

Além de serem amplamente utilizados em HAR, sensores IMUs e 1D CNNs também podem ser utilizadas na Identificação Biométrica do Usuário (BUI). Para isso, um algoritmo de cascata deve ser considerado, onde primeiramente ocorre a identificação da atividade, para então ocorrer a identificação do indivíduo.

Os sistemas de IA estão cada vez mais influenciando a vida humana, realizando

tarefas como curadoria de conteúdo online, assistência na tradução de documentos e respondendo a perguntas sobre variados assuntos (Caldwell *et al.*, 2020, Gohel *et al.*, 2021). Apesar dessas capacidades, a falta de compreensão sobre como esses sistemas tomam decisões gera uma relutância em confiar e depender deles. O conceito de "IA explicável"(XAI) refere-se a sistemas de inteligência artificial que podem elucidar o processo de tomada de decisão para os usuários (Gohel *et al.*, 2021). Modelos de ML clássicos, como árvores de decisão, são tipicamente altamente interpretáveis, mas muitas vezes não atingem o desempenho de modelos de DL, especialmente em cenários complexos onde padrões nos dados não são facilmente identificáveis. Em contrapartida, modelos de DL frequentemente falham em fornecer explicações claras sobre suas decisões. Dessa forma, observa-se um *trade-off* entre desempenho e interpretabilidade, sendo esse um desafio significativo na implementação de IA em diversas aplicações.

Ao trabalhar com dados de imagem, técnicas como *Gradient-weighted Class Activation Mapping* (grad-CAM) e de reconhecimento de objetos como *You Only Look Once* (YOLO), podem ser aplicadas para compreender as decisões dos modelos. Porém, em HAR com dados de séries temporais, a aplicação dessas técnicas ainda não é amplamente explorada. A maioria dos artigos foca apenas em métricas de desempenho para avaliar o potencial e a capacidade dos modelos, deixando uma lacuna no uso na análise qualitativa, por meio de métodos XAI em DL, especialmente em HAR, limitando a compreensão dos desenvolvedores sobre as capacidades e limitações de seus modelos.

Para avançar na interpretabilidade em HAR, técnicas como grad-CAM e *t-Distributed Stochastic Neighbor Embedding* (t-SNE) oferecem novos caminhos. O grad-CAM, utilizado em redes convolucionais, gera mapas de saliência que indicam as áreas importantes para a tomada de decisão dos modelos. Enquanto isso, o t-SNE, uma técnica inicialmente proposta para redução de dimensionalidade e compressão de informação, revela-se eficaz quando combinada com redes profundas, fornecendo visualizações detalhadas dos dados. Essas abordagens inovadoras, discutidas também por (Selvaraju *et al.*, 2019), prometem aprimorar a compreensão dos modelos de DL aplicados ao HAR.

As principais contribuições desta tese estão relacionadas à adaptação dos métodos grad-CAM e t-SNE para a analisar a tomada de decisão de 1D CNNs aplicadas ao contexto de HAR. Ao adaptar grad-CAM e t-SNE para séries temporais e modelos de aprendizado profundo, esta pesquisa propõe visualizações inovadoras que tornam possível a compreensão qualitativa dos modelos, permitindo a identificação de viés, tornando possíveis a criação de hipóteses sobre tomada de decisão dos modelos. Considerando que a combinação de métodos XAI com modelos de DL aplicados ao contexto

de HAR fornece maior confiabilidade e compreensão das 1D CNNs, essas técnicas podem ainda ser utilizadas para compor um protocolo para o desenvolvimento de modelos de HAR mais robusto, composto por uma etapa de explicações visuais com as técnicas apresentadas neste trabalho.

1.1 Motivação

HAR tem se tornado cada vez mais importante devido à crescente utilização de dispositivos vestíveis e à demanda por aplicações inteligentes capazes de monitorar e analisar os comportamentos humanos em tempo real. A aplicação prática dessas tecnologias pode ser encontrada em diversos setores, como saúde, esporte, entretenimento, segurança e automação residencial.

1D CNNs têm sido amplamente adotadas no campo de HAR, devido à sua capacidade de aprender padrões complexos nos sinais de entrada e alcançar resultados promissores em termos de classificação. No entanto, apesar do sucesso desses modelos em termos de desempenho numérico, a compreensão e explicação do processo de tomada de decisão por trás dessas arquiteturas de DL permanece um desafio.

A motivação deste trabalho reside na necessidade de abordar essa lacuna, aplicando novos métodos XAI para entender e visualizar o aprendizado e tomada de decisão das 1D CNNs aplicadas ao HAR. Através da utilização de técnicas como grad-CAM e t-SNE, buscamos fornecer *insights* sobre como esses modelos aprendem a tomar decisões e qual a qualidade das características aprendidas durante o processo de treinamento, permitindo uma análise mais aprofundada de seu funcionamento.

Esta investigação ganha importância ao fornecer visualizações que tornam possível a explicação do comportamento de arquiteturas de DL, que até então, são consideradas caixas-pretas. Com os métodos propostos, somos capazes de identificar possíveis vieses nos modelos e nos conjuntos de dados, analisar como diferentes abordagens de validação impactam o aprendizado do modelo e orientar a seleção de um modelo mais adequado com base em critérios além do desempenho numérico. Adicionalmente, a incorporação dessas técnicas explicáveis no padrão protocolo para o desenvolvimento de um modelo de HAR possui o potencial de elucidar as reais capacidades e limitações dos modelo desenvolvidos, tornando as 1D CNNs mais confiáveis e aplicáveis em cenários do mundo real.

1.2 Hipóteses da Pesquisa

Neste estudo, propomos as seguintes hipóteses relacionadas à interpretação e explicação de 1D CNNs, aplicadas ao HAR:

1. A adaptação da técnica grad-CAM para séries temporais permitirá uma melhor compreensão das regiões importantes para a tomada de decisão das 1D CNNs.
2. A adaptação da técnica t-SNE para ser utilizada em modelos de aprendizado profundo permitirá uma análise geral sobre as características aprendidas pelas redes 1D CNNs.
3. As visualizações propostas facilitarão a interpretação qualitativa dos modelos. Tornando possível a análise do comportamento dos modelos.
4. As arquiteturas de rede convolutivas apresentadas serão equiparáveis as presentes no estado da arte de aplicações HAR.
5. A combinação de métodos XAI com modelos profundos aplicados a HAR proporcionará maior confiabilidade e entendimento das 1D CNNs, propondo uma melhoria no protocolo padrão para o desenvolvimento de modelos de HAR.

1.3 Objetivos

1.3.1 Objetivos gerais

Investigar técnicas de aprendizado de máquina e reconhecimento de padrões visando a interpretação e explicação de modelos de aprendizado profundo aplicados ao reconhecimento humano de atividades.

1.3.2 Objetivos específicos

1. Propor uma adaptação da técnica de visualização grad-CAM para séries temporais e avaliar a sua utilização na análise da tomada de decisão de redes convolucionais unidimensionais no reconhecimento de atividades humanas e identificação biométrica do usuário.
2. Propor uma adaptação da técnica t-SNE para visualização da distribuição dos dados de entrada do ponto de vista das características aprendidas por redes convolucionais unidimensionais em tarefas de reconhecimento de atividades humanas.

3. Propor arquiteturas equiparáveis ao estado da arte de aplicações de HAR.

4. Investigar a utilização das técnicas propostas nos objetivos anteriores para a identificação de viés no conjunto de dados e na tomada de decisão dos modelos.

1.4 Organização da tese

No Capítulo 2, Fundamentação teórica, explicamos conceitos básicos para um melhor entendimento da tese, tais como: IA, ML, DL, XAI e conceitos relacionados a HAR. Com o Capítulo 3, Trabalhos relacionados, são apresentados abordagens comuns em HAR, BUI e XAI, é dado um foco maior nos trabalhos que utilizam as mesmas bases de dados aplicados nesta tese. Em seguida, no Capítulo 4, Metodologia, descrevemos nossa abordagem de geração de explicações visuais para modelos 1D CNNs aplicados a HAR e BUI. No Capítulo 5, Resultados, apresentamos os resultados dos modelos treinados. Em discussões, Capítulo 6, são apresentados resultados explicáveis e análise desses resultados. Na conclusão, Capítulo 7, é apresentada uma visão de conjunto da tese, bem como possíveis trabalhos futuros. Por fim, no Apêndice apresentamos os artigos científicos publicados ao longo do desenvolvimento desta tese.

2

Fundamentação Teórica

Este capítulo apresenta os conceitos básicos para o entendimento desta tese. Primeiramente, elucidamos os conceitos básicos de IA, ML e DL. Apresentamos também o conceito de XAI, explicando ainda os algoritmos t-SNE e grad-CAM para um melhor entendimento da proposta da tese. Em seguida, abrangemos o protocolo para o desenvolvimento de modelos de HAR, com uma etapa adicional que contém a melhoria proposta neste trabalho. Por fim, apresentamos um breve resumo sobre o capítulo.

2.1 Inteligência Artificial: Uma Visão Geral

IA é um ramo da ciência da computação que busca criar máquinas e sistemas capazes de executar tarefas que normalmente requerem inteligência humana (Goodfellow *et al.*, 2016). A IA pode ser dividida em várias subáreas, sendo as mais notáveis ML e o DL (Russell & Norvig, 2010).

2.2 Aprendizado de Máquina

O ML é uma técnica de IA que permite aos sistemas aprender e melhorar a partir de experiências anteriores, normalmente baseada em dados, sem a necessidade de programação explícita (Mitchell, 1997). Os algoritmos de ML são classificados em três categorias principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (Goodfellow *et al.*, 2016).

- **Aprendizado supervisionado**: Neste tipo de aprendizado, os algoritmos são treinados com um conjunto de dados rotulados e aprendem a fazer previsões ou

classificações com base em padrões observados nos dados de treinamento (Bishop, 2016).

- **Aprendizado não supervisionado**: Aqui, os algoritmos analisam dados não rotulados e identificam padrões ou estruturas ocultas, como agrupamento e redução de dimensionalidade, com algoritmos como Análise de Componentes Principais (PCA) e t-SNE (Goodfellow *et al.*, 2016).

- **Aprendizado por reforço**: Neste cenário, os algoritmos aprendem a tomar decisões baseadas em recompensas e punições ao interagir com um ambiente (Sutton & Barto, 2018).

2.2.1 Correlação de Pearson

A correlação de Pearson é uma medida estatística que quantifica o grau de relação linear entre duas variáveis contínuas (Rodgers & Nicewander, 1988). É comumente usada no aprendizado de máquina para análise exploratória de dados, seleção de variáveis e avaliação do desempenho do modelo em problemas de regressão (Bishop, 2016).

A correlação de Pearson é calculada usando a seguinte fórmula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{2.1}$$

em que, r_{xy} é o coeficiente de correlação de Pearson entre as variáveis x e y, n é o número de observações, x_i e y_i são as i-ésimas observações das variáveis x e y, respectivamente, $\overline{x}$ e $\overline{y}$ são as médias das observações de x e y, respectivamente.

A correlação de Pearson varia de -1 a 1, com os seguintes significados:

- $r_{xy} = 1$: Correlação positiva perfeita, indicando uma relação linear positiva entre as variáveis.

- $r_{xy} = -1$: Correlação negativa perfeita, indicando uma relação linear negativa entre as variáveis.

- $r_{xy} = 0$: Nenhuma correlação, sugerindo que não há relação linear entre as variáveis.

Também é possível a interpretação dos valores de correlação de Pearson intermediários. Por exemplo, uma correlação de 0.9 está próxima uma correlação positiva perfeita.

No contexto deste trabalho a correlação de Pearson será utilizada para avaliar a relação entre o alto desempenho do método de validação SD e BUI.

2.2.2 Redução de dimensionalidade: t-SNE versus PCA

A redução de dimensionalidade é um importante ramo no campo de aprendizado não supervisionado. Ela pode ser utilizada para simplificar dados e facilitar a análise, buscando manter a maior quantidade de informação possível nos dados comprimidos. Ela pode ser utilizada para produzir visualizações e na construção de modelos. Duas abordagens populares para a redução de dimensionalidade são o t-SNE e a PCA.

O PCA é uma técnica linear de redução de dimensionalidade amplamente utilizada (Jolliffe & Cadima, 2016). O algoritmo busca encontrar uma base ortogonal de componentes principais (PCs) que capturem a maior parte da variância nos dados. Os dados são então projetados nesses componentes principais, reduzindo sua dimensionalidade. A equação base do PCA é mostra abaixo:

$$\mathbf{Y}_{n\times k} = \mathbf{X}_{n\times p}\mathbf{W}_{p\times k} \tag{2.2}$$

em que, $\mathbf{X}$ é a matriz de dados de entrada, com dimensão $n \times p$, onde n é o número de observações e p é o número de variáveis. A matriz $\mathbf{W}$ é a matriz de pesos ou de transformação linear, com dimensão $p\times k$, onde k é o número de componentes principais que se deseja extrair. A matriz $\mathbf{Y}$ é a matriz de dados transformados, com dimensão $n \times k$.

Por outro lado, O t-SNE é uma técnica de redução de dimensionalidade mais rebuscada e que consegue reduzir a dimensionalidade de dados que possuem uma correlação não linear (der Maaten & Hinton, 2008). O t-SNE busca preservar a estrutura local dos dados ao mapeá-los em um espaço bidimensional ou tridimensional. A ideia principal é minimizar a divergência de Kullback-Leibler (KL) entre as distribuições de probabilidade que representam as similaridades entre os pontos de dados no espaço original e no espaço de dimensões reduzidas.

A função de custo do t-SNE é definida como:

$$C_{t-\mathrm{SNE}} = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.3}$$

A divergência de KL é uma medida da diferença entre duas distribuições de probabilidade. No algoritmo t-SNE, a divergência KL é usada como função de custo, em que P e Q representam as distribuições de probabilidade de alta e baixa dimensão, respectivamente. Na Equação (2.3), p_{ij} denota a probabilidade de que o ponto de alta dimensão i seja selecionado como vizinho do ponto j, enquanto q_{ij} é a probabilidade de que o ponto de baixa dimensão i seja selecionado como vizinho do ponto j.

Ajustando as posições dos pontos no espaço de baixa dimensão, o t-SNE busca

minimizar esta função de custo. O gradiente da função de custo em relação às coordenadas de baixa dimensão é tipicamente calculado e usado para atualizar as coordenadas usando um algoritmo de descida de gradiente.

De forma geral, t-SNE encontra mais aplicações, uma vez que consegue comprimir informação, mesmo onde não há uma correlação linear nos dados, diferentemente do algoritmo de PCA.

Neste trabalho uma das abordagens de visualização propostas envolve integrar a técnica t-SNE com uma arquitetura de rede 1D CNN.

2.3 Aprendizado Profundo

O aprendizado profundo (DL) é uma subárea do aprendizado de máquina (ML) focada em redes neurais artificiais com múltiplas camadas (Goodfellow *et al.*, 2016). As redes neurais profundas são capazes de aprender representações hierárquicas e abstratas dos dados, o que as torna especialmente úteis para tarefas complexas, como reconhecimento de imagem, tradução automática, entre outras (LeCun *et al.*, 2015). A base para o aprendizado profundo são as redes neurais. Essas redes neurais podem ser construídas em módulos, seguindo uma estrutura de camadas Abadi *et al.* (2016), LeCun *et al.* (2015).

2.3.1 Redes Neurais Artificiais

As Rede Neural Artificials (ANNs) são uma classe de modelos de aprendizado profundo que se inspiram na estrutura e no funcionamento do cérebro humano (LeCun *et al.*, 2015). As ANNs são compostas por camadas de nós ou unidades, chamadas neurônios artificiais, que são interconectadas por sinapses artificiais com pesos associados. Esses modelos são capazes de aprender representações hierárquicas de dados e de realizar tarefas complexas, como reconhecimento de padrões, classificação e regressão (Goodfellow *et al.*, 2016).

Uma ANN com apenas uma camada oculta ainda é considerada uma rede de ML tradicional. Essa arquitetura é considerada profunda quando possui mais de uma camada oculta. Neste caso sua estrutura tipicamente consiste em uma camada de entrada, várias camadas ocultas e uma camada de saída, conforme a ilustração da Figura 2.1. Os neurônios nas camadas de entrada e saída correspondem às variáveis de entrada e de saída do problema, respectivamente. As camadas ocultas são responsáveis por aprender representações abstratas dos dados e por transformar os sinais de entrada em previsões de saída.

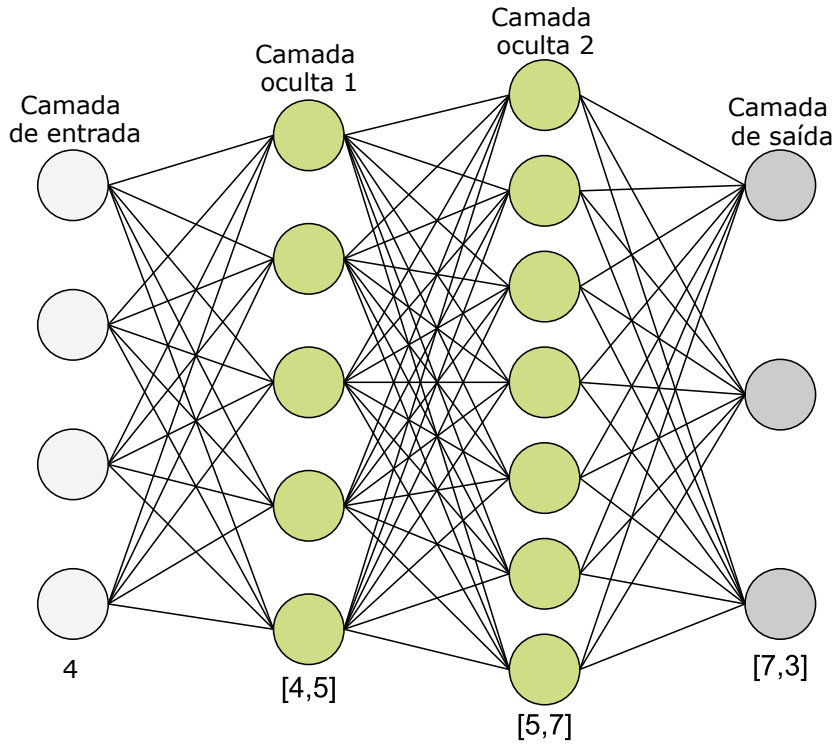


Figura 2.1. Exemplo de uma rede neural artificial profunda, com duas camadas ocultas.

Os neurônios em uma ANN processam a informação de acordo com a seguinte fórmula:

$$z_i = f\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right) \tag{2.4}$$

onde x_j são as entradas do neurônio, w_{ij} são os pesos (*weights*) das sinapses conectando a entrada j ao neurônio i, b_i é o viés (*bias*) do neurônio i, f é a função de ativação e z_i é a saída do neurônio (Goodfellow *et al.*, 2016).

As funções de ativação mais comuns incluem a função Sigmóide, a tangente hiperbólica e a função de ativação linear retificada (ReLU) (Glorot *et al.*, 2011).

O treinamento de uma ANN envolve o ajuste dos pesos e vieses das sinapses para minimizar o erro entre as previsões da rede e os valores observados. Isso é geralmente realizado usando o algoritmo de retropropagação (*backpropagation*) e a otimização de gradiente descendente (Rumelhart *et al.*, 1986). O Algoritmo 1 descreve de forma básica o algoritmo do gradiente descendente, onde a taxa de aprendizado α e o número máximo de iterações N são especificados. O algoritmo começa com uma inicialização aleatória dos parâmetros θ e, em seguida, atualiza-os iterativamente usando a derivada da função objetivo $L(\theta)$ com relação a cada parâmetro θ_i. O processo é repetido até

que o número máximo de iterações seja atingido. Ao final do algoritmo, os parâmetros ótimos θ^* são retornados.

Algoritmo 1 Gradiente Descendente

Requer Taxa de aprendizado α
Requer Número máximo de iterações N
Requer Função objetivo $L(\theta)$
Garante θ^* tal que $L(\theta^*) \leq L(\theta)$ para todos os θ
1: Inicialize θ aleatoriamente
2: Defina $t \leftarrow 0$
3: **enquanto** $t < N$ **faça**
4: Calcule o gradiente $\nabla L(\theta) = \left(\frac{\partial L}{\partial \theta_1}, \ldots, \frac{\partial L}{\partial \theta_n}\right)$
5: Atualize os parâmetros $\theta \leftarrow \theta - \alpha \nabla L(\theta)$
6: Incremente $t \leftarrow t + 1$
7: **fim enquanto**
8: **devolve** θ

As ANNs podem ser adaptadas para lidar com uma ampla variedade de problemas e arquiteturas, como Rede Neural convolucionals (CNNs) para processamento de imagens e sinais (Krizhevsky *et al.*, 2017).

2.3.2 Convolução

O processo de convolução é uma operação matemática fundamental nas CNNs e desempenha um papel crucial na extração de características dos dados de entrada, preservando as relações espaciais entre os elementos. A convolução é aplicada utilizando filtros (*kernel*) que são matrizes de pequenas dimensões e valores numéricos. O processo de convolução é ilustrado na Figura 2.2

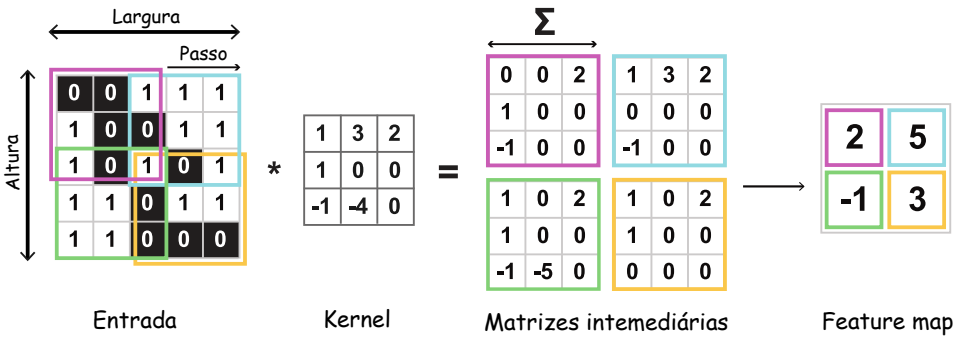


Figura 2.2. Exemplo de redes Neural artificial profunda, com duas camadas ocultas.

Inicialmente, um *kernel* de dimensões $k \times k$ é posicionado no canto superior esquerdo da matriz de entrada. Em seguida, é realizada uma operação de multiplicação

elementar entre os elementos do filtro e os elementos da matriz de entrada correspondentes. Os resultados dessas multiplicações são somados, gerando um único valor escalar. Esse valor escalar é armazenado em uma nova matriz, chamada de mapa de características (*feature map*). O filtro é então deslocado horizontalmente por um número fixo de posições, chamado de passo (*stride*). Se o filtro atingir a borda da matriz de entrada, ele será reposicionado no início da próxima linha e continuará a deslizar. O processo de multiplicação e soma é repetido para cada posição do filtro na matriz de entrada até que a convolução seja aplicada em toda a área da matriz. O resultado final é um mapa de características que representa a presença das características aprendidas pelo filtro na matriz de entrada.

Embora o processo tenha sido ilustrado para uma matriz ele pode ser aplicado de forma parecida para um sinal unidimensional. Aplicando o mesmo procedimento explicado anteriormente (Bishop, 2016).

2.3.3 Redes Neurais Convolucionais

As CNNs são uma classe especial de ANNs projetadas para processar dados com estrutura com uma relação espacial, como imagens (LeCun *et al.*, 1998). O que faz de uma ANN uma CNN é a presença da camada de convolução. Durante o treinamento da CNN, os valores dos *kernels* são ajustados para aprender características relevantes dos dados de entrada. Dessa forma, os filtros podem detectar bordas, texturas, cores e outras informações úteis para a tarefa em questão. É comum usar múltiplos filtros em uma única camada de convolução, permitindo que a rede aprenda diferentes características simultaneamente. Além disso, as CNNs geralmente têm várias camadas de convolução empilhadas, o que permite a extração de características de diferentes níveis de abstração. As primeiras camadas geralmente aprendem características de baixo nível (como bordas e texturas), enquanto as camadas mais profundas aprendem características de alto nível (como formas e padrões complexos).

Além das camadas convolutivas, as CNNs podem ser compostas por diversas outras camadas como de agrupamento (*pooling*), camadas de regularização (*dropout*), camadas completamente conectadas (*dense*), e camadas de ativação com nas ANNs. A seguir explicaremos mais detalhes de camadas profundas, focando nas camadas que foram utilizadas nas redes convolutivas do trabalho.

2.3.4 Camadas de Convolução

A camada de convolução é responsável por aprender características relevantes a partir dos dados de entrada. O processo de convolução é realizado usando filtros (ou *kernels*) que deslizam ao longo da entrada e geram um mapa de características como saída (Krizhevsky *et al.*, 2017).

2.3.5 Camadas de Agrupamento

As camadas de agrupamento (*pooling*) são usadas para reduzir a dimensionalidade espacial dos dados e ajudar a criar invariância a pequenas variações na entrada. O agrupamento mais comum é o *max pooling*, que extrai o valor máximo de uma região local da entrada, mas também existem os agrupamentos baseado em valor médio e global (Goodfellow *et al.*, 2016).

2.3.6 Camadas Densas

As camadas densas, também conhecidas como camadas totalmente conectadas, são um componente comum em redes neurais artificiais. Nelas, cada neurônio de uma camada está conectado a todos os neurônios da camada anterior e da camada seguinte. Essas camadas são responsáveis por combinar as características aprendidas nas camadas anteriores para realizar a tarefa final, como classificação ou regressão (Goodfellow *et al.*, 2016).

2.3.7 Camadas de Ativação

As camadas de ativação em uma rede neural são responsáveis por introduzir uma não-linearidade nos dados de entrada, permitindo que a rede aprenda representações mais complexas e abstratas. A função de ativação *Softmax* é geralmente usada na camada de saída de uma rede neural para tarefas de classificação multiclasse. Ela converte os valores de entrada em probabilidades normalizadas, de modo que a soma das probabilidades de todas as classes seja igual a 1 (Bishop, 2016). A função é definida como:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.5}$$

em que z_i é a entrada do neurônio i, K é o número total de classes e $\sigma(z)_i$ é a probabilidade da classe i.

Além da função *Softmax*, outras funções de ativação comuns incluem a função Sigmóide ($\sigma(x) = \frac{1}{1+e^{-x}}$), a tangente hiperbólica ($\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$) e a função de ativação linear retificada (ReLU, $f(x) = \max(0, x)$) (Glorot *et al.*, 2011).

2.3.8 *Dropout*

O *dropout* é uma técnica de regularização usada para evitar o sobreajuste (ou *overfitting*) em redes neurais (Srivastava *et al.*, 2014). Durante o treinamento, o *dropout* define aleatoriamente a saída de alguns neurônios para zero com uma probabilidade p, impedindo que esses neurônios contribuam para a etapa de treinamento atual. Isso ajuda a rede a aprender representações mais robustas e a generalizar melhor para dados não vistos.

2.4 Inteligência Artificial Explicável (XAI)

Com o avanço da IA e sua crescente integração em diversos setores da sociedade, a necessidade de compreensão e explicabilidade das decisões tomadas por esses sistemas tornou-se crucial (Gohel *et al.*, 2021, Samek *et al.*, 2017). Isso é mais crítico quando consideramos a crescente utilização de DL para solucionar problemas do cotidiano das pessoas. XAI é uma subárea da IA que busca tornar as decisões de sistemas inteligentes mais transparentes, compreensíveis e rastreáveis por humanos, auxiliando na criação de um ambiente seguro e confiável.

O uso de métodos XAI é importante para os usuários e para os desenvolvedores de modelos inteligentes. Com IA é possível atingir os seguintes objetivos (Gohel *et al.*, 2021, Samek *et al.*, 2017):

- **Compreensibilidade**: Fornecer explicações claras e intuitivas das decisões tomadas pelos sistemas de IA, permitindo compreensão sobre processos de tomada de decisão.

- **Rastreabilidade**: Estabelecer um rastreamento claro entre a entrada (dados), o processo de tomada de decisão (modelo) e a saída (decisão), permitindo a verificação e validação das decisões tomadas.

- **Capacidade**: Fornecer aos desenvolvedores a capacidade de compreensão sobre as limitações do modelo desenvolvido, ou dos dados sobre os quais esse modelo foi gerado.

2.4.1 Categorização e Comparação das Abordagens em XAI

Há duas abordagens distintas sobre XAI, transparente e pós-treinamento (Gohel *et al.*, 2021). Abordagens transparentes referem-se a modelos cuja arquitetura interna e processo de tomada de decisão são simples e facilmente compreendidos. Exemplos de modelos transparentes incluem o modelo bayesiano, árvores de decisão, regressão linear e sistemas de inferência *fuzzy*. As abordagens transparentes são benéficas quando as correlações internas de características não são particularmente complexas. Abordagens de explicabilidade pós-treinamento podem revelar o funcionamento interno e a lógica de decisão de um modelo de IA, fornecendo discernimento acerca da importância de características, conjuntos de regras, mapas de calor ou linguagem natural para ajudar os usuários a entender as informações mais relevantes e possíveis vieses (Gohel *et al.*, 2021, Samek *et al.*, 2017). Nesses casos, a técnica pós-treinamento pode fornecer uma ferramenta útil para explicar o que o modelo aprendeu, especialmente quando a relação entre os dados e as características não é simplesmente direta (Gohel *et al.*, 2021).

Dentro dos métodos pós-treinamento, temos duas abordagens distintas: modelo agnóstico e modelo específico (Gohel *et al.*, 2021). Abordagens específicas de modelo fornecem limitações de explicabilidade em relação ao algoritmo de aprendizado e à estrutura interna de um determinado modelo de DL. Em contraste, abordagens agnósticas analisam as entradas e previsões do modelo em pares para compreender os mecanismos de aprendizado e gerar explicações.

Abordagens modelo-agnósticas, como *Local Interpretable Model-agnostic Explanations* (LIME) e *SHapley additive explanations* (SHAP), podem ser usadas com características artesanais e ML clássico para determinar a importância das características de entrada (Datta *et al.*, 2016, Lipovetsky & Conklin, 2001, Ribeiro *et al.*, 2016).

Abordagens específicas de modelo são usadas em XAI para explicar as decisões tomadas por um modelo específico de aprendizado de máquina. Essas abordagens são adaptadas à arquitetura, parâmetros e processo de treinamento do modelo e geralmente dependem do conhecimento do funcionamento interno do modelo. Exemplos de abordagens específicas de modelo incluem indução de árvore de decisão, extração de regras, métodos baseados em gradiente e propagação de relevância em camadas (Gilpin *et al.*, 2018, Lu *et al.*, 2021, Selvaraju *et al.*, 2019, Zhang *et al.*, 2018).

Essas abordagens fornecem discernimentos valiosos sobre o processo de tomada de decisão do modelo e podem ajudar a melhorar o desempenho do modelo e reduzir vieses.

2.4.2 *Gradient-Weighted Class Activation Mapping*

O grad-CAM é uma abordagem específica de modelo que se enquadra na categoria de explicabilidade pós-treinamento. A técnica de mapeamento de ativação de classe ponderada por gradiente (grad-CAM) é a mais utilizada e citada hoje (Palatnik de Sousa *et al.*, 2021). Essa técnica usa um gradiente de uma classe alvo, fluindo para a última camada convolucional da arquitetura, para criar um mapa de localização que destaca as regiões da imagem que mais influenciaram a tomada de decisão do modelo. Este método fornece uma visão intuitiva do aprendizado da rede, permitindo aos usuários visualizar e entender quais características foram mais relevantes para a tomada de decisão do modelo.

Grad-CAM usa um gradiente de uma classe alvo, fluindo para a última camada convolucional da arquitetura, para criar um mapa de localização grosseiro que destaca as regiões da imagem que mais influenciaram a tomada de decisão do modelo (Palatnik de Sousa *et al.*, 2021). É sabido que as camadas convolucionais mais profundas podem abstrair informações mais complexas, então a última camada convolucional deve conter a abstração máxima do sinal (Selvaraju *et al.*, 2019). Visualizar essa informação nos dá uma ideia mais clara do que a inteligência aprendeu e como ela toma as suas decisões (Selvaraju *et al.*, 2019). Esta visualização se torna mais interessante quando é possível focar apenas nas informações que levaram a rede a associar uma entrada específica com uma classe alvo com a ajuda de mapas de localização. Grad-CAM surgiu como uma melhoria no trabalho de Zhou *et al.* (2016), que propôs mapeamento de ativação de classe (CAM) para identificar regiões discriminativas usadas por classes restritas para classificação de imagens com classificadores CNN. No entanto, a técnica CAM enfrenta uma limitação, pois só é aplicável para redes que não possuem camadas totalmente conectadas. Grad-CAM é uma generalização do CAM, tornando possível aplicá-lo a diferentes famílias de CNNs, incluindo CNNs com camadas totalmente conectadas, com saídas estruturadas, usadas em tarefas com entradas multimodais e aprendizado por reforço.

A técnica grad-CAM recebe um sinal de destino e uma classe de interesse como entrada. Em seguida, o gradiente é calculado para a classe alvo y^c em relação aos mapas de características da última camada convolucional A_{ij}^k. Conforme descrito na Equação 2.6, para obter os pesos de importância de cada neurônio α_k^c, esses gradientes obtidos são passados por uma função de agrupamento médio global.

$$\alpha_k^c = \underbrace{(1/Z \sum_i \sum_j)}_{\text{global average pooling}} \underbrace{(\partial y^c / \partial A_{ij}^k)}_{\text{gradientes via backpropagation}} \tag{2.6}$$

Em que Z representa o número total de elementos nos mapas de características. Os pesos α_k^c representam uma linearização parcial da rede profunda em relação a A, e capturam a "importância"de um mapa de características para uma classe alvo. Em seguida, calculamos uma combinação ponderada dos mapas de ativação direta, seguida por uma função ReLU para eliminar contribuições negativas. Com isso, obtemos o mapa de calor das regiões de importância em nossa rede $L_{Grad-CAM}^c$, de acordo com a Equação 2.7.

$$L_{Grad-CAM}^c = \text{ReLU} \underbrace{(\sum_k \alpha_k^c A^k)}_{\text{combinação linear}} \tag{2.7}$$

2.5 Reconhecimento Humano de Atividades

A IA é a base de qualquer tecnologia usada para o HAR (Gupta *et al.*, 2022, Hayat *et al.*, 2022). Os sistemas de HAR identificam as atividades humanas usando dados capturados por dispositivos vestíveis ou smartphones (Ferrari *et al.*, 2021, Zhang *et al.*, 2022). O principal objetivo do HAR é melhorar os resultados de saúde para pessoas que sofrem de doenças crônicas, como diabetes, doença de Parkinson e até mesmo idosos (Cvetković *et al.*, 2016, Hayat *et al.*, 2022, Kazemimoghadam & Fey, 2022). O HAR possui inúmeras aplicações no setor de saúde e bem-estar para pessoas que desejam monitorar sua saúde e aptidão física (Ferrari *et al.*, 2021, Zhang *et al.*, 2022).

Diversos sensores podem ser utilizados para implementar o HAR. Um sensor relevante nesse domínio é o acelerômetro. O acelerômetro pode detectar movimentos de alta e baixa frequência (Ferrari *et al.*, 2021, Gu *et al.*, 2021). Um giroscópio e um magnetômetro podem ser usados para integrar os dados do acelerômetro (Ferrari *et al.*, 2021, Zhang *et al.*, 2022). Um exemplo de sinal do acelerômetro utilizado para tarefas de HAR pode ser visto na Figura 2.3.

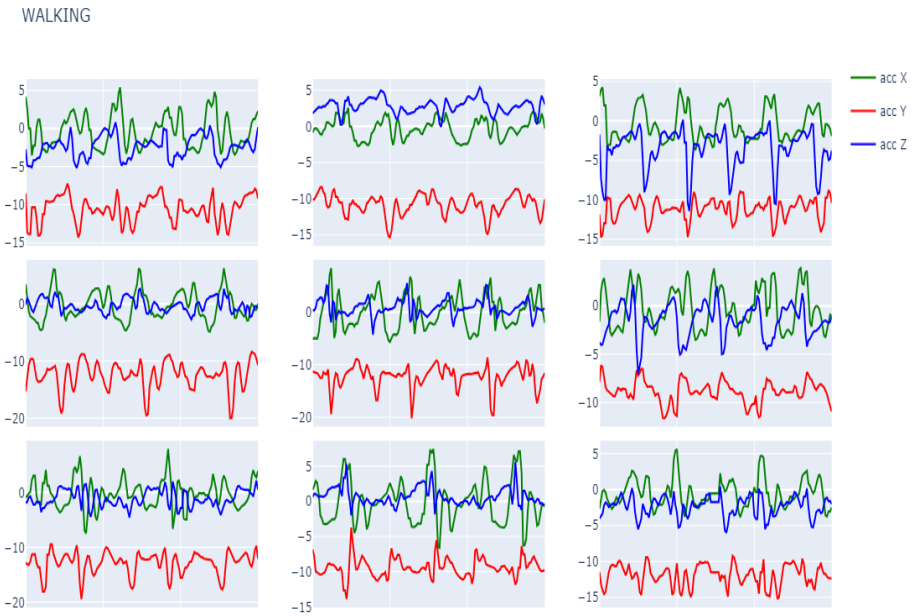


Figura 2.3. Exemplo de sinais temporais utilizados para classificar HAR. No exemplo todas as amostras foram de caminhada. As amostras foram obtidas do bando UniMiB-SHAR

Quanto mais informações sensoriais forem fornecidas a um sistema de IA, mais fácil será aprender um padrão. No entanto, em aplicações práticas, mais sensores significam mais recursos computacionais, que geralmente não estão disponíveis em dispositivos com capacidades limitadas, como *smartwatches* ou *smartphones*. Portanto, desenvolver uma solução que utilize apenas um tipo de sensor torna-se mais adequado para uso em cenários do mundo real.

2.5.1 Protocolo para Reconhecimento de Atividades Humanas

A maioria das aplicações de HAR utiliza um protocolo padrão para o desenvolvimento desses sistemas (Bragança *et al.*, 2022, Ferrari *et al.*, 2021, Zhang *et al.*, 2022). No entanto, esse protocolo pode sofrer alterações dependendo do problema específico. Geralmente, o ARP consiste em seis etapas: aquisição de dados, pré-processamento, segmentação, extração de características e classificação e avaliação. Embora esse protocolo tem sido amplamente utilizados por muitos trabalhos recentes (Bragança *et al.*, 2022, Cvetković *et al.*, 2016, Zhang *et al.*, 2022).

2.5.1.1 Aquisição

Na etapa de aquisição do HAR, é importante selecionar os sensores apropriados para capturar os dados necessários. Além dos acelerômetros, outros sensores como giros-

cópios, magnetômetros, barômetros e sensores de frequência cardíaca podem ser utilizados para obter informações adicionais sobre as atividades humanas (Ferrari *et al.*, 2021, Gupta *et al.*, 2022, Zhang *et al.*, 2022).

É fundamental garantir que os dados sejam coletados de forma confiável e consistente. Isso pode envolver o desenvolvimento de aplicativos ou sistemas embarcados que sigam protocolos de aquisição estabelecidos, bem como a definição de critérios para a qualidade dos dados obtidos (Micucci *et al.*, 2017, Reyes-Ortiz *et al.*, 2016). Em alguns casos, o uso de câmeras ou microfones pode ser necessário para rotular as atividades e enriquecer os dados capturados pelos sensores (Micucci *et al.*, 2017, Reyes-Ortiz *et al.*, 2016).

A frequência de amostragem dos sensores, como o acelerômetro, é um aspecto importante a ser considerado. A taxa de amostragem de 50 Hz é comumente utilizada na literatura, mas pode variar dependendo da aplicação e das características das atividades a serem reconhecidas (Ferrari *et al.*, 2021). Um cuidado especial deve ser tomado para evitar o excesso de amostragem, que pode levar a um consumo excessivo de energia e de recursos computacionais (Aquino *et al.*, 2022).

2.5.1.2 Pré-processamento

O pré-processamento dos dados é uma etapa crítica para garantir a qualidade e a utilidade dos dados no reconhecimento de atividades. Os sensores podem realizar um pré-processamento básico no nível do hardware, como a aplicação de filtros para reduzir o ruído ou a normalização dos dados (Ferrari *et al.*, 2021). No entanto, é comum realizar um pré-processamento adicional após a aquisição para melhorar a qualidade dos dados e remover artefatos indesejados.

O pré-processamento pode envolver técnicas de processamento de sinais digitais, como filtragem, suavização, detecção de picos e extração de características relevantes (Ferrari *et al.*, 2021). Além disso, técnicas de aprendizado de máquina, como normalização dos dados, seleção de características e redução de dimensionalidade, podem ser aplicadas para otimizar o desempenho dos algoritmos de reconhecimento de atividades (Aquino *et al.*, 2022).

É importante ressaltar que o pré-processamento deve ser cuidadosamente avaliado, pois certas transformações nos dados podem afetar as características originais e, consequentemente, a capacidade do modelo de reconhecimento de atividades de aprender e generalizar corretamente (Ferrari *et al.*, 2021).

2.5.1.3 Segmentação

A segmentação dos dados é uma etapa essencial no processo de reconhecimento de atividades. Consiste em dividir os dados contínuos em segmentos menores chamados janelas, que contêm informações relevantes sobre as atividades a serem reconhecidas (Ferrari *et al.*, 2021).

A definição do tamanho adequado da janela é crucial para capturar as características temporais das atividades. Um tamanho de janela muito pequeno pode não fornecer informações suficientes, enquanto um tamanho muito grande pode suavizar as características distintivas das atividades. Na literatura de HAR, um tamanho de janela comumente utilizado é de aproximadamente três segundos, mas pode variar dependendo do contexto e das características específicas das atividades (Ferrari *et al.*, 2021).

Existem duas abordagens principais para segmentação: janelamento definido por eventos e janelamento de deslizamento. No janelamento definido por eventos, as atividades são identificadas e delimitadas por eventos específicos, como o início e o fim de uma atividade (Ferrari *et al.*, 2021). Já no janelamento de deslizamento, as janelas são deslizadas ao longo do tempo com um certo intervalo de sobreposição entre elas (Ferrari *et al.*, 2021). A sobreposição entre as janelas ajuda a capturar informações contínuas e suavizar possíveis variações na duração das atividades (Ferrari *et al.*, 2021).

Além disso, a segmentação dos dados deve ser acompanhada pela rotulagem adequada das amostras. Cada janela segmentada deve ser rotulada com a atividade correspondente para a criação do conjunto de dados de treinamento e teste (Aquino *et al.*, 2022). É importante garantir uma distribuição equilibrada de amostras entre as diferentes classes de atividades, de modo a evitar viés e permitir que o modelo aprenda de forma adequada todas as atividades de interesse (Reyes-Ortiz *et al.*, 2016, Shoaib *et al.*, 2015).

2.5.1.4 Extração de Características

A extração de características desempenha um papel fundamental em HAR, pois envolve a transformação de dados brutos em um conjunto reduzido de características que possam ser processadas de forma mais eficiente por algoritmos de aprendizado de máquina (Ferrari *et al.*, 2021, Gupta *et al.*, 2022). Existem duas abordagens principais para a extração de características: características construídas manualmente e características aprendidas (Ferrari *et al.*, 2021, Gupta *et al.*, 2022). Os métodos de características construídas manualmente empregam relações matemáticas determinadas por especialistas no assunto, enquanto as características aprendidas são obtidas por meio de algoritmos

de aprendizado de máquina e correlações entre os dados. Embora o aprendizado profundo seja amplamente utilizado em outras áreas para a extração de características, no contexto do HAR, muitos estudos ainda se baseiam em características construídas manualmente devido às limitações de processamento dos dispositivos móveis quando o mesmo é utilizado para inferência dos modelos. As características construídas manualmente podem ser obtidas por meio de atributos estatísticos no domínio temporal, transformada de Fourier no domínio da frequência e discretização para a obtenção de características simbólicas. Por outro lado, as características aprendidas podem ser extraídas por meio de redes neurais profundas, como as redes neurais convolucionais (CNNs) ou os autoencoders (Ferrari *et al.*, 2021). As CNNs extraem informações por meio da operação de convolução, enquanto os autoencoders comprimem os dados para extrair informações de alto nível em seu espaço latente. Nas CNNs, as características aprendidas são representadas pelos coeficientes do kernel, enquanto nos autoencoders, elas correspondem ao espaço latente da conversão.

2.5.1.5 Classificação

Sistemas de classificação podem ser abordados por meio de um paradigma orientado a modelo ou orientado a dados (Ferrari *et al.*, 2021). A abordagem orientada a modelo busca reproduzir manualmente o funcionamento do sistema físico, empregando regras de composição que podem descrever o problema como uma equação. Métodos orientados a dados utilizam ML para descobrir correlações nos dados e são mais comumente usados em HAR (Ferrari *et al.*, 2021). Algoritmos de ML e DL utilizam métodos estatísticos e matemáticos para desenvolver algoritmos capazes de resolver problemas de classificação. Classificadores tradicionais de ML, como *Naive Bayes*, *Support Vector Machines* e árvores de decisão, requerem extração de características, uma vez que não podem lidar diretamente com dados brutos. Em contraste, sistemas de DL podem lidar com tarefas simples e complexas, mas requerem uma grande quantidade de dados. A escolha do algoritmo pode ter um impacto significativo no desempenho da classificação (Aquino *et al.*, 2022). Após a seleção da arquitetura do classificador, ele pode ser treinado em Python usando *frameworks* populares, como TensorFlow ou PyTorch para DL, ou Scikit-learn para ML tradicional (Abadi *et al.*, 2016, Aquino *et al.*, 2022).

2.5.1.6 Avaliação

Após selecionar um algoritmo de classificação para resolver um determinado problema, a próxima etapa é escolher uma técnica de validação adequada para avaliar o desempenho do modelo. Essa etapa é crucial para garantir que o modelo seja capaz de

generalizar a tarefa, ou seja, de realizar previsões precisas em dados não vistos anteriormente.

As estratégias de validação dependentes do sujeito (SD) e independentes do sujeito (SI) são duas abordagens comumente utilizadas na área de aprendizado de máquina (Ferrari *et al.*, 2021). A estratégia SI é caracterizada pelo fato de não utilizar dados do usuário final durante o treinamento do modelo. Nessa abordagem, o conjunto de dados é dividido em subconjuntos de treinamento e teste de forma aleatória, sem levar em consideração as informações específicas do usuário. Essa abordagem é útil quando se deseja avaliar a capacidade do modelo de generalizar a tarefa para diferentes usuários, sem depender de características individuais.

Por outro lado, a abordagem SD aproveita informações específicas do usuário durante o treinamento do modelo. Nesse caso, o conjunto de dados é dividido em subconjuntos de treinamento e teste de maneira a preservar a dependência entre os dados de um mesmo usuário (Ferrari *et al.*, 2021). Essa estratégia é particularmente relevante quando existem diferenças significativas entre os usuários, como diferentes estilos de interação, preferências ou características individuais que possam afetar o desempenho do modelo. Ao utilizar a estratégia SD, o modelo tem a oportunidade de aprender com os dados de cada usuário individualmente, levando em consideração suas particularidades.

Uma técnica comumente utilizada para a validação de modelos é o método *hold-out* (Aquino *et al.*, 2022). Nesse método, o conjunto de dados é dividido em dois subconjuntos mutuamente exclusivos: um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento é utilizado para ajustar os parâmetros do modelo, enquanto o conjunto de teste é utilizado para avaliar o desempenho do modelo em dados não vistos anteriormente. A proporção em que o conjunto de dados é dividido pode variar, sendo comum utilizar uma divisão de 70% para treinamento e 30% para teste, por exemplo.

A Figura 2.4 apresenta um diagrama elucidando as diferenças entre as estratégias de validação SD e SI.

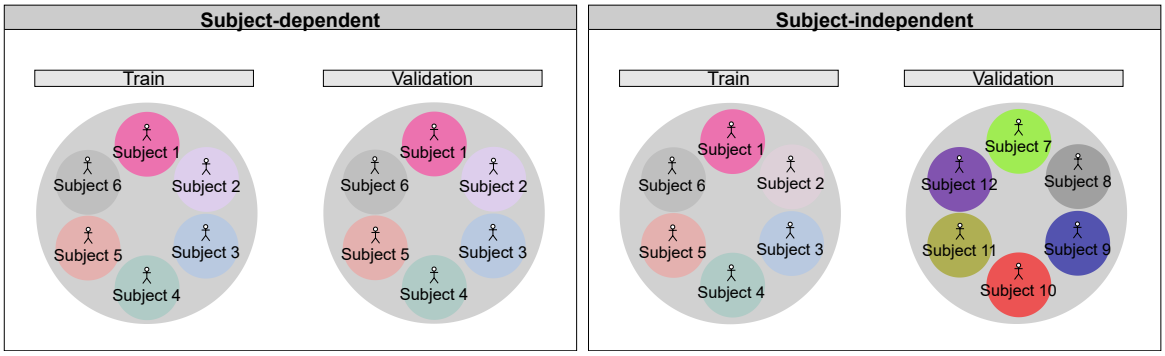


Figura 2.4. Estratégias de validação sujeito dependente (SD) e sujeito independente (SI). Onde em SD dados dos mesmos indivíduos estão no treino e validação, enquanto em SI há dados de indivíduos diferentes no treino e validação. Adaptada de (Aquino *et al.*, 2022)

O desempenho do modelo criado durante o treinamento e a validação deve ser avaliado usando um conjunto de métricas, conforme mostrado na Tabela 2.1. A acurácia, a sensibilidade, a precisão e o F1-score são as medidas de avaliação mais comumente utilizadas em HAR (Ferrari *et al.*, 2021).

Verdadeiro Positivo (TP), *Verdadeiro Negativo* (TN), *Falso Positivo* (FP) e *Falso Negativo* (FN) são os elementos essenciais de todas as métricas. O conceito de TP, TN, FP e FN é baseado na categorização binária. Com uma técnica de um-contra-todos, eles podem ser estendidos para classificação multiclasse. Nesse conceito, a classe alvo é considerada positiva, enquanto todas as outras classes são combinadas em uma classe negativa. A métrica mais popular, acurácia, reflete a proporção de assertividade geral. Embora a acurácia seja uma métrica fácil de compreender, ela não reflete o desempenho real do classificador quando temos um conjunto de dados com um desequilíbrio entre as classes. A métrica de precisão ilustra como o algoritmo lida com a acurácia ao prever amostras positivas. Se tivermos uma alta precisão, nosso modelo reconhece efetivamente as amostras TP. A sensibilidade mede como nosso modelo lida com os falsos negativos. Essa métrica é aplicada quando há uma preocupação significativa com FN. A precisão e a sensibilidade são usadas para calcular a F1-score. Essa medida é crucial quando há uma distribuição desequilibrada das classes.

Tabela 2.1. Métricas mais comumente usadas em sistemas HAR.

Métrica	Equação
Acurácia	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensibilidade	$\frac{TP}{TP+FN}$
Precisão	$\frac{TP}{TP+FP}$
F1-score	$2 \times \frac{(\text{Precisão} \times \text{Sensibilidade})}{\text{Precisão}+\text{Sensibilidade}}$

2.6 Considerações Finais

Neste capítulo, foram apresentados os principais conceitos que estão relacionados a este trabalho. Inicialmente, foram elucidados os conceitos básicos de IA, ML e DL, os quais são fundamentais para a compreensão desta tese. A IA foi apresentada como um ramo de estudo da ciência da computação, sendo o ML um dos seus campos de estudo. O ML, por sua vez, abrange diversas áreas, das quais serão utilizadas neste trabalho o aprendizado supervisionado e o não supervisionado. Além disso, foram abordados conceitos importantes dentro do ML, como a correlação de Pearson e a redução de dimensionalidade.

No âmbito do aprendizado profundo, foram apresentados conceitos relacionados a redes neurais, destacando as principais camadas utilizadas em arquiteturas de redes convolucionais. Em seguida, abordamos a Inteligência Artificial Explicável, apresentando e categorizando diferentes métodos para a geração de explicações em modelos de ML ou DL, entre os quais se destaca o grad-CAM. Por fim, no campo do reconhecimento de atividades, foram abordados conceitos básicos e o protocolo padrão de ARP.

No próximo capítulo, serão apresentados trabalhos relacionados, no qual serão abordadas publicações anteriores que utilizaram esses conceitos no contexto do HAR.

3

Trabalhos relacionados

Este capítulo apresenta a revisão da literatura sobre trabalhos que aplicam HAR e XAI, com foco em abordagens envolvendo redes convolutivas. Nós dividimos esta seção em: (i) Aplicações e *baselines* em HAR, (ii) Identificação biométrica do usuário utilizando HAR, (iii) Abordagens XAI aplicadas a series temporais e (iv) Considerações finais.

3.1 Aplicações e *Baselines* em HAR

O campo do HAR tem suas raízes na década de 1990, com Foerster *et al.* (1999) sendo um dos primeiros a explorar o conceito. Desde então, o avanço tecnológico permitiu a introdução de diversas técnicas e algoritmos inteligentes para aprimorar os sistemas de HAR. Dentre os dispositivos utilizados, podemos citar câmeras, sensores piezoelétricos, GPS, microfones e IMUs como alguns dos mais relevantes (Ferrari *et al.*, 2021).

No contexto atual, as CNNs ganharam destaque no HAR devido à sua capacidade de invariância de escala e translação (Zhang *et al.*, 2022). A estrutura típica de uma CNN envolve uma sequência de codificação, na qual os dados são progressivamente reduzidos em dimensão à medida que se aprofundam na rede. Esse processo facilita a extração de significados semânticos de alto nível. Em comparação a métodos manuais, algumas arquiteturas de CNN demonstraram uma capacidade superior na extração de características representativas (Sani *et al.*, 2017). Uma característica distintiva das CNNs é sua flexibilidade em lidar com entradas de dados. Por exemplo, uma CNN pode ser alimentada com um sinal de 9 dimensões originado de um acelerômetro triaxial, giroscópio triaxial e magnetômetro triaxial. Além de dados brutos, as CNNs também podem processar características geradas manualmente ou aprendidas por outras redes neurais profundas (Ferrari *et al.*, 2021, Sani *et al.*, 2017).

Neste cenário, Teng *et al.* (2021) introduziram o HAR-Net, uma arquitetura que aceita características manuais como entrada. Em outra contribuição significativa, Sikder *et al.* (2019) elaboraram um modelo CNN de aprendizado supervisionado para distinguir ações humanas. Este estudo se concentrou em abordar os desafios apresentados pela natureza ciclo estacionária dos sinais de atividade física no HAR. Essa natureza implica em padrões que surgem periodicamente, mas sem um intervalo fixo. Para superar isso, os autores desenvolveram uma CNN bicameral (com duas entradas) que incorpora características de frequência e potência dos sinais de atividade humana. Ao ser avaliada no conjunto de dados UCI HAR, a CNN demonstrou uma precisão de 95,25%. Em uma abordagem semelhante, Ronao & Cho (2016) projetaram uma CNN que processa dados de 6 dimensões, originários de um acelerômetro triaxial e um giroscópio triaxial, e alcançou uma precisão de 94%.

Ao focar no campo de trabalhos que utilizam series temporais para realizar o reconhecimento das atividades físicas há muitas opções de base de dados público disponíveis. Dentre os mais populares *datasets* estão: HAPT, UniMiB-SHAR, SHO, e WISDM (Ferrari *et al.*, 2021, Micucci *et al.*, 2017, Reyes-Ortiz *et al.*, 2016, Shoaib *et al.*, 2014).

3.1.1 Baselines com o UniMiB-SHAR

O UniMiB-SHAR é composto principalmente por dados de mulheres com idades entre 18 e 60 anos, com altura entre 160 cm e 190 cm e peso entre 50 kg e 82 kg (Micucci *et al.*, 2017). Os dados foram coletados usando um dispositivo Samsung Galaxy Nexus 19250 com sistema operacional Android 5.1.1, utilizando o sensor de aceleração Bosh BMA220. A frequência de amostragem foi de 50 Hz, com uma janela de tempo de 3 segundos, resultando em 151 amostras para cada um dos 3 eixos do acelerômetro. O processo de janelamento foi baseado em eventos, utilizando a detecção de picos. Cada atividade foi realizada entre 2 e 6 vezes. Foram consideradas duas posições para o smartphone. Metade dos participantes colocou o smartphone no bolso esquerdo, enquanto a outra metade o colocou no bolso direito Micucci *et al.* (2017). Trinta indivíduos participaram dos experimentos para a criação da base. Foram realizadas 17 atividades físicas, entre elas 9 tipos de atividades diárias (ADLs) e 8 tipos de quedas.

Vários estudos recentes empregaram a base de dados UniMiB. A Tabela 3.1 oferece um resumo dos métodos e resultados dessas pesquisas relacionadas a este conjunto de dados. É importante ressaltar que todos esses estudos se basearam em 17 classes e utilizaram exclusivamente dados brutos do acelerômetro como entrada para o classificador.

Tabela 3.1. Estado da arte e *baselines* para o conjunto de dados UniMiB-SHAR. $N \times M$—N é o tamanho de entrada e M é a dimensão de entrada; SD—estratégia de validação dependente do sujeito; SI—estratégia de validação independente do sujeito; CV— técnica de validação cruzada; LOSO—estratégia de validação *Leave-one-subject-out*

Autor	Técnica de modelo	Entrada ($N \times M$)	Estratégia de validação	Resultado
Micucci *et al.* (2017)	KNN e SVM	51×3 e 151×3	SD: CV-5 e SI: LOSO	SD com 151×3— 0,830 de MAA * com KNN
Mukherjee *et al.* (2020)	*Ensemble* de CNNs	151×3	SD: CV-5	SD: 0,926 de acurácia
Tang *et al.* (2020)	CNN	151×3	SD: *Hold-out* 70/30	SD: 0,775 de F1-score ponderada
Serrão *et al.* (2021)	GRU e CNN-2D	151×3	SD: CV-5 e SI: LOSO	SD: 0,955 de MAA e 0,978 de acurácia com GRU
				SI: 0,714 de MAA e 0.798 de acurácia com CNN-2D
Lv *et al.* (2020)	ConvLSTM	151×3	SD: CV-5 and SI: LOSO	SD: 0,953 de MAA e 0,973 de F1-score ponderada
				SI: 0,784 de F1-score ponderada
Teng *et al.* (2021)	Resnet (CNN)	151×3	SD: *Hold-out* 70/30	SD: 0,806 de F1-score ponderada e 0,809 de acurácia
Chen *et al.* (2021)	CNN	151×3 e 151×1 (magnitude)	SD: Hold-out 70/30	SD: 151×3: 0,886 de acurácia
				SD: 151×1: 0,7731 de acurácia

** MAA = Mean Average Accuracy.*

Vários estudos têm se voltado para o desenvolvimento de arquiteturas inovadoras de redes neurais e metodologias de processamento, frequentemente demonstrando desempenho superior em comparação com os *benchmarks* estabelecidos (Lv *et al.*, 2020, Micucci *et al.*, 2017, Mukherjee *et al.*, 2020, Serrão *et al.*, 2021). Tais estudos comumente empregam o UniMiB-SHAR como referência, realizando uma análise comparativa com pesquisas precedentes ou evidenciando a eficácia de sua técnica em múltiplos datasets. Adicionalmente, há trabalhos que introduzem avanços nos algoritmos de otimização empregados no treinamento de redes profundas ou em sua configuração arquitetônica, contrapondo o desempenho destes métodos com abordagens mais convencionais (Cheng *et al.*, 2022, Tang *et al.*, 2020, Teng *et al.*, 2021).

Mukherjee *et al.* (2020) delinearam três modelos classificatórios: CNN-Net, Encoded-Net e CNN-LSTM. Estes modelos são fundamentados em CNNs unidimensionais. A entrada é uma matriz bidimensional derivada da aplicação de janelamento com sobreposição no sinal. O paradigma adotado consiste em cada rede neural fornecer uma predição, sendo a classe majoritariamente predita a selecionada. Em contraste, Micucci *et al.* (2017) utilizaram dados brutos como entrada para os algoritmos KNN e SVM. Esta pesquisa é particularmente relevante por ser o *benchmark* inicial, visto que os

autores conceberam o dataset UniMiB-SHAR. A motivação central do estudo foi introduzir o conjunto de dados e demonstrar que, mediante uma implementação simplista, é viável efetuar reconhecimento de atividades através de algoritmos de aprendizado de máquina.

Tang *et al.* (2020) propuseram uma CNN eficiente do ponto de vista computacional, implementando filtros análogos a blocos de Lego. Tal abordagem visa contornar as restrições de empregar aprendizado profundo em dispositivos de processamento limitado, como *wearables*. Adotando filtros compactos, os autores corroboraram a possibilidade de minimizar os gastos computacionais e de memória ao empregar CNNs. Esta rede é descrita pelos autores como ágil, concisa e acurada.

Por sua vez, Lv *et al.* (2020) delinearam uma rede híbrida de DL baseada na arquitetura ConvLSTM. Esta abordagem amalgama duas arquiteturas profundas, CNN e LSTM, objetivando capitalizar as informações obtidas pela CNN no domínio temporal. Complementarmente, um módulo de interconexão densa foi inserido para aprimorar a comunicação inter camadas e um módulo de agregação de características de diversas camadas foi implementado para exaustivamente explorar características em domínios espaciais e temporais. Os pesquisadores ainda agregam características oriundas de cada camada convolucional, ponderando sua relevância em distintos segmentos do sinal.

Teng *et al.* (2021) avançaram tanto na metodologia de treinamento quanto na arquitetura de redes neurais residuais. Estas redes, uma subclassificação específica de CNNs, possuem uma estrutura não sequencial, incorporando informações de camadas antecedentes. O protocolo de treinamento por blocos adota funções de perda locais para aplicações HAR. Divergindo da retropropagação global padrão, perdas locais baseadas em entropia cruzada e similaridade supervisionada são empregadas para treinar individualmente cada bloco residual. Tal abordagem culmina em aprimoramentos na precisão classificatória e eficiência de memória durante o treinamento.

Finalmente, Chen *et al.* (2021) objetivaram aprimorar as operações da CNN mediante a convolução condicionalmente parametrizada, focando no HAR em tempo real em dispositivos móveis e vestíveis.

3.1.2 Baselines com as bases HAPT e SHO

A Tabela 3.2 demonstra as utilizações frequentes das bases de dados SHO e HAPT em trabalhos recentes. A pesquisa sobre o uso da base de dados HAPT revela que alguns autores optam por agrupar as classes de Transição Postural (PT) do conjunto de dados. Essa estratégia de agrupamento pode envolver dividir os PTs em um único subgrupo Reyes-Ortiz *et al.* (2016), Thu & Han (2020) ou dois subgrupos Thu & Han

(2020, 2021). No entanto, há apenas um número limitado de estudos que tentam classificar os PTs em 12 classes distintas.

Em relação aos trabalhos que utilizaram a base de dados HAPT, Reyes-Ortiz *et al.* (2016) a introduziram combinando-a com uma base de dados existente que consistia principalmente de transições posicionais. Os autores desenvolveram um sistema que emprega um classificador de máquina de vetor de suporte e um conjunto de regras heurísticas para classificar seis tipos de atividades humanas. Em seu artigo, eles agruparam todas as atividades de transição postural em uma única classe, demonstrando que essa estratégia melhorou o desempenho de reconhecimento das seis Atividades da Vida Diária (ADLs) consideradas. Thu & Han (2021) criaram a estrutura HiHAR, que emprega duas arquiteturas profundas, Redes Neurais Convolucionais (CNNs) e Redes de Memória de Longo e Curto Prazo Bidirecionais (BiLSTM), para classificar atividades PT em dois subgrupos Thu & Han (2020). Thu & Han (2020) avaliaram suas arquiteturas considerando PTs como um subgrupo, dois subgrupos e como atividades separadas.

Em seu trabalho sobre o conjunto de dados SHO, Jiang & Yin (2015) propôs uma estrutura que utilizava dados de acelerômetro, giroscópio e sensores de aceleração linear. Os dados temporais foram convertidos em uma imagem por meio de um processo que utilizava a Transformada Discreta de Fourier (DFT). A estrutura empregava as oito atividades disponíveis no SHO, mas só utilizava dados de um sensor colocado no pulso. Por outro lado, Bragança *et al.* (2022) consideraram todas as posições de sensores disponíveis, mas apenas seis atividades. As atividades foram selecionadas para serem compatíveis com as atividades físicas usadas em outros conjuntos de dados em seu estudo.

De forma abrangente, a evolução e a crescente sofisticação das técnicas HAR nos últimos anos evidenciam uma demanda contínua e uma necessidade intrínseca de incorporar tecnologias de detecção de atividades humanas. A incorporação e aprimoramento das CNNs no campo do HAR transcenderam as expectativas ao melhorar significativamente a precisão do reconhecimento. Elas também ilustram a versatilidade e a capacidade de adaptação desses modelos para processar variadas tipologias e complexidades de dados. Além disso, a consolidação de bases de dados como UniMiB-SHAR, HAPT e SHO, estabelecendo-se como *benchmarks* essenciais, enfatiza a relevância de contar com padrões uniformizados. Esses *benchmarks* são cruciais para a avaliação e comparação objetiva de inovações metodológicas e técnicas emergentes na área.

Tabela 3.2. Comparação de desempenho de diferentes modelos HAR usando os conjuntos de dados HAPT e SHO. $N \times M$—N é o tamanho da entrada e M é a dimensão da entrada; FE representa a extração de Características; SD—estratégia de validação dependente do sujeito; SI—estratégia de validação independente do sujeito; CV—técnica de validação cruzada; LOSO—estratégia de validação *Leave-one-subject-out*.

Autor	Técnica de modelo	Entrada ($N \times M$)	Estratégia de validação	Classes	Resultado
Reyes-Ortiz *et al.* (2016)	Sistema TAHAR com SVM	561 × 1 vetor de FE (Acc + Gyro)	Sem subconjunto de validação	6 BA + 1 PT	0,9700 de precisão
Thu & Han (2021)	HiHAR-8 e CNN	128 × 6 (Acc + Gyro)	SI: 24/6	6 BA + 2 PT	0,9798 de precisão com HiHAR-8
					0,9440 de precisão com CNN
Thu & Han (2020)	BiLSTM e CNN	FE:* DWT (Acc + Gyro)	SD: *Hold-out* 80/20	6 BA + 1 PT	0,9634 de precisão com BiLSTM
				6 BA + 6 PT	0,9487 de precisão com BiLSTM
					0,9171 de precisão com CNN
Jiang & Yin (2015)	CNN 2D	36 × 68 - imagem de ** DFT (Acc + Gyro + Aceleração linear)	SD: *Hold-out* 70/30	8 BA	0,9993 de precisão
Bragança *et al.* (2022)	Floresta Aleatória	64 × 1 FE (Acc)	SD: 10-CV, SI: LOSO	6 BA	SD: 0,98 de MAA, SI: 0,8412 de MAA

**DWT = Transformada de Onda Discreta; **DFT = Transformada Discreta de Fourier.*

3.2 Identificação biométrica do usuário utilizando HAR

Os sistemas de identificação biométrica podem ser categorizados em duas principais vertentes (Mekruksavanich & Jitpattanakul, 2021). A primeira é baseada em características estáticas, que compreendem aspectos físicos como o rosto, impressões digitais, entre outros. Já a segunda é baseada em características dinâmicas, voltada para as particularidades comportamentais, como sinais eletrocardiográficos, voz ou uma atividade específica. Muitos estudos têm explorado a possibilidade de utilizar os sensores integrados em smartphones para reconhecer atividades dos usuários. Contudo, poucos avançaram no potencial deste tipo de dado para a BUI (Mekruksavanich & Jitpattanakul, 2021).

Mekruksavanich & Jitpattanakul (2021) sugerem a identificação do usuário por meio de diferentes sinais de atividade física, utilizando um acelerômetro triaxial e um giroscópio triaxial. Seu modelo primeiramente reconhece a atividade do usuário e, em seguida, realiza a autenticação. Há uma rede de autenticação para cada atividade física catalogada na base de dados. Os autores empregaram uma CNN e uma LSTM, alcançando acurácias de 91,77% e 92,43%, respectivamente. Os conjuntos de dados

utilizados foram o conjunto de reconhecimento de atividade humana UCI e o conjunto de reconhecimento de atividade humana USC HAD. O primeiro possui dados de 30 indivíduos executando seis atividades diárias, enquanto o segundo contém dados de atividades de 14 indivíduos, sendo 7 homens e 7 mulheres, que realizam 12 atividades diárias. O tamanho da janela foi de 128 amostras com frequência de amostragem de 50 Hz em ambos os conjuntos.

Juefei-Xu *et al.* (2012) propuseram um sistema de identificação de usuários baseado na detecção prévia de diferentes ritmos de caminhada. Os eventos de caminhada são categorizados em velocidades como normal e rápida. O desempenho foi avaliado em três cenários. No primeiro, que considerou apenas amostras de caminhada normal, os autores atingiram uma Taxa de Verificação (TV) de 99,4%, uma Taxa de aceitação falsa (TAF) de 0,1% e uma acurácia de 95%. No segundo cenário, para rápida versus rápida, obtiveram 96,8% para TV, 0,1% para TAF, e 98% de acurácia. No terceiro, utilizando todas as amostras, o desempenho caiu para 66,1% para TV, 0,1% para TAF e 40% de acurácia. TAF é uma métrica que mede o número médio de aceitações falsas em um sistema biométrico, também chamada de taxa de falso negativo. Por outro lado, a TV mede o número médio de aceitações positivas previstas em relação ao *ground truth*. Um acelerômetro triaxial e um giroscópio triaxial foram utilizados. Para a extração de características, foram empregados métodos de extração manual (*hand-crafted*). O tamanho da janela usado foi de 1 segundo. O estudo ressaltou que a velocidade de caminhada do usuário deve ser levada em consideração para uma identificação correta.

Gamble & Huang (2020) apresentaram uma abordagem de aprendizado profundo para HAR, focando na "assinatura comportamental"do indivíduo, representada pelas características extraídas dos sinais do acelerômetro e giroscópio do smartphone. Utilizando um modelo de rede neural convolucional unidimensional (1D CNN), a pesquisa buscou determinar se é possível não apenas identificar a atividade realizada, mas também a identidade do indivíduo que a executa, com base nos sinais mencionados. Em experimentos iniciais no conjunto de dados MotionSense, o modelo 1D-CNN para classificação de atividades obteve uma acurácia de 96,77%, enquanto o modelo de identificação alcançou 82,37% em uma janela de tempo de apenas um segundo. Contudo, é ressaltado que a decisão de desenvolvimento de usar uma duração de amostra de um segundo pode ter limitado a eficácia dos modelos, sugerindo possíveis tópicos de pesquisa para aprimorar a acurácia em estudos futuros.

Em resumo, a identificação biométrica através da Análise do HAR usando sensores de *smartphones* ou *smartwatches* emerge como uma abordagem promissora, abarcando tanto características estáticas quanto dinâmicas do indivíduo. Estudos demonstraram a eficácia de algoritmos em distinguir atividades e identificar usuários com acurácias

impressionantes. A utilização de acelerômetros e giroscópios, em particular, tem demonstrado ser eficaz na captura de "assinaturas comportamentais"que são únicas para cada indivíduo. A constante evolução das técnicas e algoritmos sugere um futuro onde os sistemas de HAR poderão desempenhar um papel fundamental na segurança e autenticação de usuários em ambientes digitais.

3.3 Abordagens XAI em séries temporais

A ascensão da inteligência artificial (IA) nos últimos anos transformou a paisagem de diversas indústrias e aplicações. Nesse cenário emergente, a demanda por sistemas de IA explicável (XAI - *Explainable AI*) tornou-se mais necessária. Estes sistemas visam oferecer justificativas claras e transparentes para suas decisões, permitindo que usuários e desenvolvedores entendam e confiem nas ações automatizadas. A aplicação de XAI em séries temporais se torna imprescindível, já que a interpretação de padrões e tendências mutáveis ao longo do tempo é desafiadora (Schlegel *et al.*, 2019, Viton *et al.*, 2020). Tais dados, advindos de variados sensores, fornecem um vasto volume de informações que, se isolados de um contexto apropriado, podem parecer intrincados (Schlegel *et al.*, 2019).

No mundo das séries temporais, as técnicas de XAI mostram-se não só úteis, mas indispensáveis. Elas têm o potencial de esclarecer tendências complexas e nuances que se modificam continuamente. Várias abordagens têm sido exploradas, incluindo métodos baseados em gradiente, estrutura e substitutos. Cada uma dessas técnicas possui suas peculiaridades e aplicabilidades, fornecendo distintos *insights* sobre as decisões tomadas pelos modelos de IA (Samek *et al.*, 2017, Schlegel & Keim, 2021). Por exemplo, enquanto os métodos de gradiente destacam-se pela agilidade computacional para análises de amostras específicas, os métodos estruturais exploram a arquitetura interna das redes neurais, avaliando pesos e viéses para deduzir uma pontuação da saída à entrada. Por outro lado, abordagens como Local Interpretable Model-agnostic Explanations (LIME) e SHapley Additive exPlanations (SHAP) utilizam uma estratégia de perturbação, gerando novas instâncias baseadas na amostra de entrada e avaliando suas pontuações.

Ao abordar técnicas visuais, Assaf & Schumann (2019) propõem uma abordagem inovadora para visualizar séries temporais. A ideia é representar cada ponto no tempo como um retângulo, variando a cor conforme a relevância. Isso permite uma visualização intuitiva dos momentos mais críticos. Já Selvaraju *et al.* (2019) oferecem uma perspectiva diferente com a técnica *grad*, otimizada para a visualização de camadas

convolutivas de CNNs. A despeito de sua inovação, Viton *et al.* (2020) identificaram lacunas nessa técnica, observando padrões que podem ser confusos. Assim, eles introduziram modificações para melhorar a clareza e a interpretabilidade das visualizações.

Embora a pesquisa em técnicas explicáveis tenha avançado, especialmente para reconhecimento de atividades a partir de dados sensoriais, ainda existem desafios significativos. Bettini *et al.* (2021) focam nessa questão, ressaltando a complexidade de entender o raciocínio intrínseco dos modelos ao classificar atividades humanas. O *framework* XAR por eles proposto busca solucionar esse problema, integrando técnicas explicáveis a modelos tradicionais como florestas aleatórias e máquinas de vetores de suporte.

Outras abordagens, como a de Atzmueller *et al.* (2018), escolhem classificadores baseados em regras. Estes encapsulam relações claras entre eventos sensoriais e ações, tornando a lógica do modelo transparente e acessível. A proposta de Arrotta *et al.* (2022) é ainda mais ambiciosa: com o DeXAR, um *framework* avançado, buscam aprimorar o reconhecimento de atividades por meio da conversão de dados sensoriais em imagens semânticas, abrindo caminho para o uso de técnicas explicáveis em duas dimensões.

Em resumo, enquanto a IA continua avançando e se integrando a aplicações práticas, a necessidade de compreensão e transparência acompanha esse crescimento. As técnicas de XAI surgem como uma resposta a esse desafio, e, no domínio das séries temporais, sua importância é ainda mais pronunciada. A busca contínua é por métodos que não só explicam, mas também aprimoram os sistemas de IA, tornando-os mais confiáveis e eficazes. Sem considerar o potencial que as explicações proveniente das aplicações das técnicas XAI tem, ao quebrar paradigmas já conhecidos na área, como por exemplo, como a escolha do método de validação impacta no aprendizado do modelo.

3.4 Considerações finais

Ao longo deste capítulo, explorou-se o cenário atual da pesquisa em HAR e XAI, ilustrando uma diversidade em estudos, métodos e implementações. Como vimos anteriormente, os domínios de HAR e XAI necessitam de abordagens explicáveis para o reconhecimento de atividades humanas.

As Redes Neurais Convolucionais (CNNs), em particular, emergem como um marco significativo no domínio do HAR, oferecendo melhorias notáveis em termos de precisão e versatilidade, especialmente ao se deparar com variados e complexos con-

juntos de dados. Também ressaltamos a importância de base de dados públicas como a base UniMiB-SHAR, exemplificando o papel vital dos *benchmarks* padronizados na realização de comparações entre diferentes estratégias.

Discutimos ainda a vertente promissora da identificação biométrica do usuário via HAR, onde algoritmos não só identificam a atividade em andamento, mas também a pessoa específica que a realiza. Esta característica, identificada através de sensores comuns em aparelhos móveis, pode ser crucial em futuros mecanismos de segurança e verificação.

No campo de XAI, identificou-se uma demanda crescente por sistemas de IA mais claros e entendíveis, principalmente aplicados a modelos profundo, especialmente quando trata-se de análise de séries temporais. Com a integração cada vez maior destas tecnologias no cotidiano, entender e justificar as ações de sistemas de IA torna-se importante.

Por fim, este capítulo apresentou um panorama de HAR e XAI repleto de dinamismo e potencial. Há um cruzamento evidente entre precisão, singularidade e clareza, o que norteará os rumos futuros destes campos de estudo.

No próximo Capítulo, será apresentada a metodologia utilizada no trabalho, a qual servirá de base para entendimento dos experimentos executados nesta tese.

4

Metodologia

Este capítulo detalha a metodologia adotada para o desenvolvimento da pesquisa, cujo foco reside em propor visualizações que ampliem a explicabilidade de modelos com camadas 1D CNN no reconhecimento humano de atividades utilizando dados sensoriais. A pesquisa pode ser classificada quanto à sua natureza, objetivos metodológicos e procedimentos técnicos, conforme descrevemos a seguir.

A natureza deste trabalho é exploratória e aplicada, visto que busca explorar métodos de visualização como meio de interpretar os resultados gerados por modelos complexos de aprendizado de máquina, aplicando-os no contexto específico do reconhecimento de atividades humanas através de dados sensoriais. Este estudo se posiciona como qualitativo e quantitativo, uma vez que se preocupa tanto com a interpretação das relações internas dos modelos quanto com a mensuração e análise estatística dos resultados.

Nós dividimos este capítulo em: (i) *Datasets*, (ii) Ambiente de desenvolvimento (iii) Arquiteturas, (iv) Visualização utilizando o método grad-CAMM (v) Visualização utilizando o método t-SNE e (vi) *Framework* proposto.

4.1 Conjunto de dados

Este seção aborda análises nos conjuntos de dados públicos utilizados no trabalho, são eles: UniMiB-SHAR, HAPT e SHO. O primeiro foi utilizado para explorar a capacidade do método grad-CAM. Os conjuntos HAPT e SHO foram utilizados para a aplicação do método XAI com t-SNE.

4.1.1 UniMib-SHAR

A base de dados utilizado nesse trabalho foi o UniMiB-SHAR (Micucci *et al.*, 2017). O qual foi previamente descrito na Seção 3.1. Nesta base de dados 30 indivíduos realizaram 17 atividades físicas, dentre as quais 9 eram consideradas ADL e 8 eram diferentes tipos de quedas. O conjunto tinha um total de 11.771 amostras de acelerômetro bruto, coletados a 50 Hz por 3 segundos.

Para um melhor entendimento sobre a base de dados, a Tabela 4.1 mostra a distribuição de amostras presentes nesta base. Como mostrado na coluna *N° de amostras*, o dataset é desbalanceado. Finalmente, através dessa análise exploratória inicial é possível ver que nem todos os indivíduos realizaram todas as atividades físicas.

Tabela 4.1. Distribuição dos dados do conjunto de dados UniMiB-SHAR.

Ação	Rótulo	Descrição da atividade	N° de amostras	N° indivíduos
ADL	*StandingUpFS*	Levantando de sentado	153	24
	StandingUpFL	Levantando de deitado	216	28
	Walking	Caminhando	1738	29
	Running	Correndo	1985	30
	GoingUpS	Subindo escadas	921	30
	Jumping	Pulando	746	30
	GoingDownS	Descendo escadas	1324	30
	LyingDownFS	Deitando de sentado	296	29
	SittingDown	Sentando	200	28
Fall	*FallingForw*	Caindo de frente	529	30
	FallingRight	Caindo a direita	511	30
	FallingBack	Caindo para trás	526	30
	HittingObs	Desviando de obstáculo	661	30
	FallingPS	Caindo com proteção	484	30
	FallingSC	Caindo para trás de uma cadeira	434	30
	Syncope	Síncope	513	30
	FallingLeft	Caindo a esquerda	534	30

Para HAR foram obtidos dois conjuntos de dados, por meio da implementação das estratégias de validação de dados SD e SI, vide a Seção 2.5.1.6. Para a estratégia SD, os dados foram divididos de maneira aleatória, utilizando o número 42 como semente

de estado aleatório, por meio da biblioteca Sklearn em Python. A proporção entre treinamento e validação foi de 70/30. Quanto à estratégia SI, trabalhos relacionados frequentemente empregam 1 sujeito para validação. Tal abordagem é conhecida como LOSO. Em nossa pesquisa, visando manter a mesma proporção de dados usada na estratégia dependente do sujeito e assegurar uma comparação equivalente entre ambas abordagens, optamos por utilizar os sujeitos de 1 a 9 para validação e os 21 sujeitos restantes para treinamento, visto que 9 representa 30% do número total de indivíduos.

Para realizar a identificação biométrica do usuário (BUI), os dados foram primeiramente filtrados por atividade, na qual cada atividade física era considerada um subconjunto. Exemplificando, para as amostras da classe *Walking*, todas as amostras disponíveis de todos os usuários que realizaram a atividade foram consideradas. Após isso, os dados foram divididos em treino e validação considerando a proporção 70/30 e a semente randômica 42. A Figura 4.1 ilustra o método de validação e divisão de dados implementado para obter o dataset de BUI.

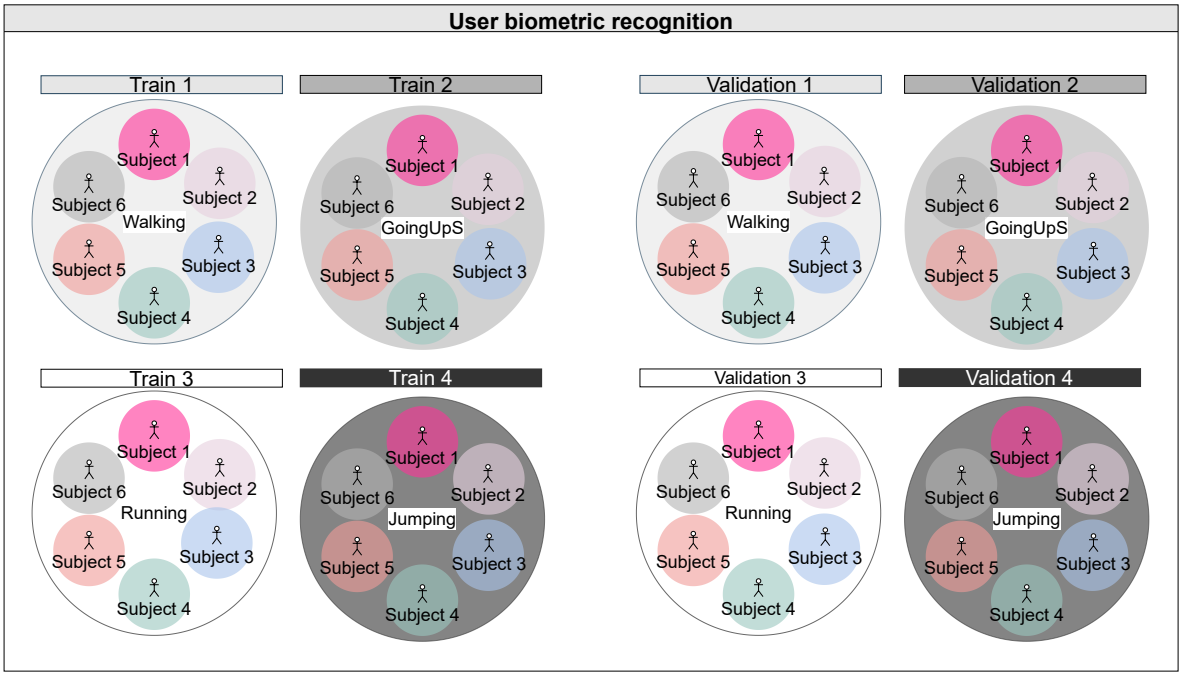


Figura 4.1. Método de divisão dos dados aplicados para realizar a identificação biométrica do usuário (BUI). Os círculos maiores representam subconjuntos separados, enquanto os círculos menores representam os indivíduos. Os dados de cada atividade física foram treinados e validados de forma separada. No mundo real, esse sistema precisaria primeiramente do reconhecimento da atividade para então depois realizar a identificação do indivíduo. Adaptada de (Aquino *et al.*, 2022).

4.1.2 SHO

Os autores coletaram dados de sete atividades físicas: caminhar, correr, sentar, ficar de pé, trotar, andar de bicicleta, subir escadas e descer escadas (Shoaib *et al.*, 2014). Dez voluntários realizaram cada tarefa por 3–4 minutos como parte do projeto de captura de dados. Todos os dez sujeitos eram homens entre 25 e 30 anos. Exceto pelo ciclismo, os estudos foram conduzidos dentro de uma instalação universitária. Cada um desses participantes estava equipado com cinco smartphones para uso em cinco diferentes posturas corporais: bolso direito, bolso esquerdo, posição no cinto voltada para a perna direita usando um clipe de cinto, braço superior direito, pulso direito.

Para os testes, foram utilizados smartphones Samsung Galaxy SII (i9100). Para cada movimento, os dados foram coletados à taxa de 50 amostras por segundo para todos os cinco locais simultaneamente. Foram coletadas informações de acelerômetro, giroscópio, magnetômetro e sensor de aceleração linear (Shoaib *et al.*, 2014).

Neste trabalho, para a geração do conjunto de dados, foi utilizada uma janela de 3 segundos, com 50% de sobreposição. Os dados foram coletados apenas da posição da cintura usando um cinto, e apenas o sensor de acelerômetro foi considerado. Para o método dependente do sujeito (SD), implementamos uma estratégia de particionamento aleatório com embaralhamento, aderindo a uma divisão convencional de 70/30 para treinamento e validação. Adicionalmente, para a abordagem independente do sujeito (SI), esforçamo-nos para manter a mesma proporção na partição, resultando em 7 sujeitos para treinamento e 3 sujeitos para validação. A validação foi realizada usando os sujeitos 1, 2 e 3.

SHO é um conjunto de dados altamente balanceado. Para nossa configuração, foi obtido um total de 4130 amostras, com exatamente 590 amostras por classe, e todos os dez sujeitos forneceram o mesmo número de amostras.

4.1.3 HAPT

A base de dados de Reconhecimento de Atividade Humana Usando Smartphones da UCI foi expandido neste conjunto de dados (Reyes-Ortiz *et al.*, 2016). Em vez dos sinais pré-processados dos sensores de smartphones que foram fornecidos na versão 1, esta versão ofereceu os sinais inerciais brutos originais dos sensores. Além disso, os rótulos de atividade foram revisados para incorporar mudanças posturais que estavam ausentes da versão anterior do conjunto de dados.

O conjunto de dados público HAPT foi obtido de trinta participantes com idades entre 19 e 48 anos (Reyes-Ortiz *et al.*, 2016). O conjunto de dados continha sinais inerciais brutos obtidos de sensores de aceleração linear de 3 eixos e velocidade angular

de 3 eixos integrados em um smartphone equipado na cintura pelo usuário. Ele incluía 6 atividades básicas (BAs): em pé, sentado, deitado, caminhando, subindo escadas e descendo escadas, e 6 transições posturais (PTs) entre três posturas estáticas, de pé para sentado, sentado para de pé, sentado para deitado, deitado para sentado, de pé para deitado e deitado para de pé.

Os dados foram coletados a uma frequência de amostragem constante de 50 Hz. Os sinais foram então sincronizados com as gravações do experimento para que pudessem ser usados como verdade fundamental para a rotulagem manual (Reyes-Ortiz *et al.*, 2016).

Para a geração do conjunto de dados, o mesmo procedimento usado no conjunto de dados SHO foi aplicado, ou seja, uma janela de tempo de 3 segundos com 50% de sobreposição, e apenas um sensor de acelerômetro. Os dados foram coletados apenas da posição da cintura usando um cinto. A Tabela 4.2 exibe mais detalhes sobre a distribuição dos dados do conjunto de dados. Como mostrado na coluna N° Amostras, o HAPT é um conjunto de dados desbalanceado. Finalmente, como mostrado na coluna N° sujeitos, podemos ver que nem todos os participantes realizaram todas as atividades, especialmente para PT.

Para o método dependente do sujeito (SD), implementamos uma estratégia de particionamento aleatório com embaralhamento, aderindo a uma divisão convencional de 70/30 para treinamento e validação. Na abordagem SI, levamos em conta que nem todos os indivíduos realizaram todas as atividades físicas. Portanto, para o subconjunto de validação, os primeiros nove indivíduos, que realizaram todas as atividades, foram escolhidos: sujeitos 1, 2, 3, 4, 5, 6, 13, 17 e 18. Os sujeitos restantes foram usados no subconjunto de treinamento.

Tabela 4.2. Classes de atividades e transições posturais (PT) no conjunto de dados HAPT com rótulos correspondentes, número de amostras e número de sujeitos.

Ação	Rótulo	N° Amostras	N° Sujeitos
BA	Em pé	855	30
	Sentado	782	30
	Deitado	851	30
	Caminhando	746	30
	Subindo escadas	687	30
	Descendo escadas	614	30
PT	De pé para sentado	39	24
	Sentado para de pé	14	10
	Sentado para deitado	59	30
	Deitado para sentado	51	28
	De pé para deitado	71	29
	Deitado para de pé	50	29

4.2 Ambiente de Desenvolvimento

Os experimentos realizados neste trabalho foram conduzidos em um ambiente de desenvolvimento especificamente configurado para atender às demandas computacionais dos modelos de aprendizado de máquina empregados. A base desse ambiente foi um computador laptop modelo Dell G3, equipado com uma placa de vídeo NVIDIA RTX 3050 e um processador AMD Ryzen da série 5000, operando sob o sistema operacional Windows 11. Essas especificações garantiram o equilíbrio necessário entre poder de processamento e eficiência energética, permitindo treinamentos e execuções de modelos complexos de maneira eficaz.

Para treinar as arquiteturas convolutivas e executar os experimentos de obtenção das visualizações com t-SNE e grad-CAM, foi necessária a configuração de um ambiente de desenvolvimento que incluísse ferramentas e bibliotecas específicas para aprendizado profundo. O software fundamental para a execução destes experimentos incluiu:

- **Python 3.9:** A linguagem de programação escolhida, devido à sua ampla adoção na comunidade científica e ao suporte robusto para bibliotecas de aprendizado de máquina e processamento de dados.

- **TensorFlow e Keras:** Para a implementação e treinamento das arquiteturas

convolutivas, utilizou-se TensorFlow, juntamente com sua API de alto nível Keras, aproveitando sua flexibilidade e facilidade de uso para prototipagem rápida.

- **Scikit-learn:** Para as operações de pré-processamento de dados, assim como para a aplicação e análise do t-SNE, foi utilizada a biblioteca Scikit-learn, que oferece uma vasta gama de ferramentas estatísticas e algoritmos de aprendizado de máquina.

- **Matplotlib e Plotly:** Para a geração de visualizações, incluindo as geradas pelo grad-CAM e pelos mapas do t-SNE, recorreu-se às bibliotecas Matplotlib e Plotly, que permitem a criação de gráficos estatísticos de alta qualidade.

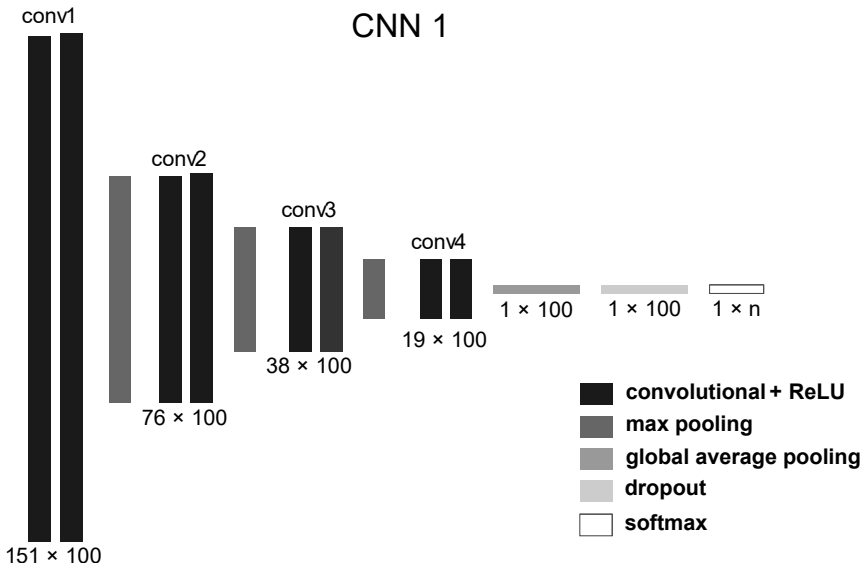
4.3 Arquiteturas

As simulações foram conduzidas utilizando as arquiteturas de redes convolucionais CNN1 e CNN2, mostradas na Figura 4.2. Essas arquiteturas foram propostas para essa tese e se basearam em outros trabalhos na área de reconhecimento de séries temporais com dados brutos e processamento de imagens (Howard *et al.*, 2017, Simonyan & Zisserman, 2015, Srivastava *et al.*, 2014). Por exemplo, a decisão de empilhar dois blocos de camadas convolutivas e uma camada de *max-pooling* foi baseada na rede VGGNet (Simonyan & Zisserman, 2015). A VGGNet apresenta arquiteturas que fazem uso extensivo de blocos com duas camadas convolucionais seguidas por uma camada *max-pooling*. O valor de *dropout* de $0,5$ foi escolhido baseado na rede AlexNet (Srivastava *et al.*, 2014). A decisão de não utilizar camadas densas no final da rede foi baseada na rede MobileNet, que são uma classe de modelos eficientes projetados para dispositivos móveis e aplicações embarcadas com restrições de processamento e memória (Howard *et al.*, 2017).

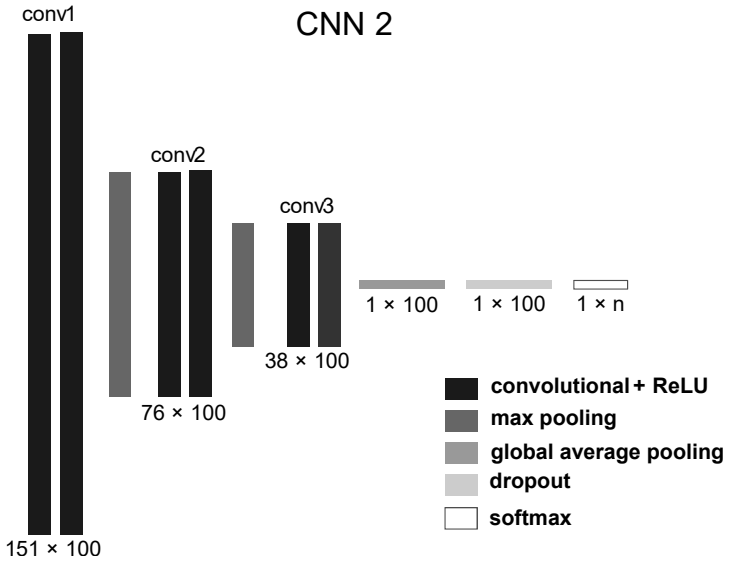
Os modelos foram treinados e implementados usando a linguagem Python e o *framework* TensorFlow 2 (Abadi *et al.*, 2016). Inicialmente, destacamos as semelhanças entre os dois modelos. Em ambos os casos, o otimizador Adam foi utilizado com seus valores de hiperparâmetros padrão. Cada modelo foi submetido a 300 épocas de treinamento. Cada bloco convolucional continha duas camadas, cada uma com 100 mapas de características. A função de ativação ReLU foi empregada. Em cada camada de *Max-pooling*, o *poolsize* e o *step* foram definidos como 2. Para a camada de *dropout*, foi aplicado um valor de $0,5$. A camada Softmax continha n neurônios, onde n representa o número de classes do problema. Um método de *callback* personalizado foi utilizado para criar *checkpoints* após cada época de treinamento, selecionando o

modelo ótimo com base na métrica macro F1-score para o conjunto de validação. O tamanho do *batchsize* foi definido como 256. Ambas as arquiteturas receberam sinais idênticos de 151×3 em suas entradas.

Em seguida, discutimos as diferenças entre as duas arquiteturas. A primeira arquitetura, CNN1, compreende quatro blocos convolucionais, cada um com duas camadas convolucionais. Em contraste, a segunda arquitetura, CNN2, incorpora apenas três blocos convolucionais. Além disso, o tamanho do *kernel* da CNN1 é 8, enquanto o da CNN2 é 4.



(a) Arquitetura CNN1 com 4 blocos convolucionais



(b) Arquitetura CNN2 com 3 blocos convolucionais

Figura 4.2. As arquiteturas CNN implementadas com TensorFlow 2. Adaptada de (Aquino *et al.*, 2022)

4.4 Visualização utilizando o método grad-CAM

Grad-CAM é geralmente aplicado a imagens. Usaremos esta técnica para uma série temporal de dados do acelerômetro. Para este propósito, fizemos algumas modificações no método original descrito na Seção 2.4.2.

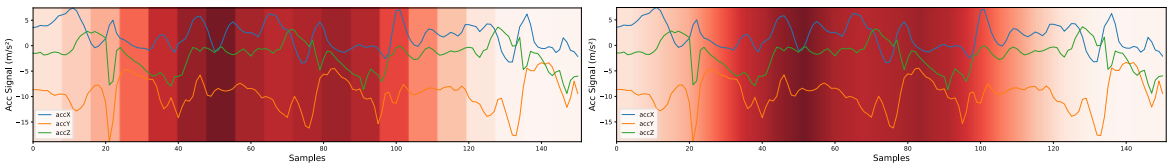
Em dados temporais, não precisamos calcular o agrupamento médio global em

duas dimensões, apenas em uma dimensão, como mostrado na seguinte Equação 4.1:

$$\alpha_k^c = \underbrace{(1/Z \sum_i)}_{\textit{global average pooling}} \underbrace{(\partial y^c / \partial A_{ij}^k)}_{\text{gradientes via } \textit{backpropagation}} \tag{4.1}$$

onde α_k^c representa os pesos ponderados do mapa de características da camada k para a classe c, servindo como um indicativo escalar da importância global de um determinado mapa de características na previsão da classe c. O termo $\frac{1}{Z}\sum_i$ corresponde ao *Global Average Pooling*, onde Z é o número total de elementos no mapa de características e a operação de somatório, $\sum_i$, representa a média dos valores no mapa de características. Por fim, $\frac{\partial y^c}{\partial A_{ij}^k}$ denota a derivada parcial da saída da classe c, y^c, em relação ao mapa de características A_{ij}^k na camada k. Este último termo captura a sensibilidade da saída da classe a pequenas variações em cada posição específica do mapa de características, obtida através da retropropagação (*backpropagation*).

Para o mapa de calor, não usamos a função ReLU. Normalizamos o sinal entre 0 e 1. Ao plotar o mapa de calor, usamos o método de interpolação de vizinho mais próximo para interpolar o sinal ao mesmo tamanho que o sinal de entrada. Interpolação linear é atualmente a mais utilizada com grad-CAM. No entanto, ao trabalhar com sinais, o resultado não é claro o suficiente quando usamos este método. A Figura 4.3 mostra que se estamos interessados em ver bem as regiões de importância, o resultado é mais evidente quando aplicamos o método de interpolação de vizinho mais próximo, pois as bordas são mais evidentes, enquanto no método de interpolação bilinear, essas regiões de importância parecem ser mais suaves, tornando mais difícil distinguir uma área muito importante de uma com importância intermediária.



(a) Interpolação pelo vizinho mais próximo. **(b)** Interpolação Bilinear.

Figura 4.3. Comparação entre os métodos de interpolação por vizinho mais próximo e bilinear. O método de interpolação por vizinho mais próximo deixa mais explicita as regiões importantes para uma série temporal, em especial para sinais do acelerômetro triaxial . Adaptada de (Aquino *et al.*, 2022).

A técnica proposta foi utilizada após o treinamento dos modelos. Para obter a visualização, uma amostra de uma classe alvo é escolhida, analisa-se a predição que a rede faz com relação a essa amostra e passa-se para o algoritmo de grad-CAM. O

algoritmo então gera um vetor de importância, do mesmo tamanho da dimensão da informação na última camada convolutiva da CNN. Essa informação é normalizada entre 0 e 1. Após isso utilizamos o algoritmo de interpolação por vizinho mais próximo para obter o sinal com a mesma dimensão de entrada.

Esse processo permite obter um gráfico de calor, no qual podemos observar os trechos do sinal que mais influenciaram na tomada de decisão do modelo. Para melhorar a qualidade da visualização é aplicado um limiar na informação do grad-CAM. Por exemplo, em alguns casos a média da saída do grad-CAM é $0,5$, mas os trechos de informação que mais influenciaram tem um valor acima de $0,7$, dessa forma, todos os valores abaixo de um limiar estabelecido são levados para 0, e os demais valores permanecem com o seu valor original. Todo o processo é resumido no Algoritmo 2 a seguir.

Algoritmo 2 Processo de Visualização com grad-CAM

Requer Amostra x da classe alvo
Requer Modelo CNN treinado
Garante Gráfico de calor do sinal influente
1: Faça a predição com a amostra x usando o modelo
2: Obtenha o vetor de importância usando grad-CAM
3: Normalize o vetor de importância para o intervalo [0,1]
4: Use interpolação por vizinho mais próximo para coincidir com a dimensão de entrada
5: Estabeleça um limiar T (e.g., $T = 0,7$)
6: **para** cada valor v no vetor de importância $\leftarrow$
7: **se até então** $v < T$ **faça**
8: Defina $v = 0$
9: **fim se**
10: **fim para**
11: **devolve** Gráfico de calor do vetor de importância modificado

4.5 Visualização utilizando o método t-SNE

T-SNE é frequentemente adotado como uma alternativa ao PCA para redução de dimensionalidade, conforme discutido na Seção 2.2.2. Em contextos de séries temporais, o método é comumente aplicado ao sinal de entrada para uma exploração inicial dos dados.

Neste estudo, apresentamos uma abordagem inovadora no contexto de séries temporais, empregando o t-SNE em uma das camadas de uma rede profunda convolucional. Possibilitando a visualização da qualidade das características extraídas a partir do

conjunto de dados. Os modelos baseados em CNN são treinados sob supervisão para classificar atividades. Após o treinamento, os dados são inseridos no modelo. Cada amostra é processada até alcançar a penúltima camada da rede, imediatamente antes do estágio de classificação. Essa representação intermediária (que podemos chamar de *embeddings*, ou características aprendidas) é então fornecida ao algoritmo t-SNE, que reduz a dimensionalidade de cada amostra para duas dimensões. O resultado é um conjunto de dados bidimensional, cuja visualização permite analisar a distinção entre as classes. O processo pode ser resumido no Algoritmo 3.

Algoritmo 3 Processo para obter a visualização com t-SNE

1: Treinar a rede profunda de forma supervisionada para HAR.
2: **para** cada amostra no conjunto de dados ← 0 **até** n **faça**
3: Propagar a amostra através da rede até a penúltima camada.
4: Extrair os *embeddings* (características aprendidas) da penúltima camada.
5: Armazenar o rótulo e a classe predita, para utilizar durante a plotagem do gráfico.
6: **fim para**
7: Aplicar o algoritmo t-SNE ao novo conjunto de dados feito a partir dos *embeddings*, para reduzir a dimensionalidade para duas dimensões.
8: Plotar o conjunto de dados bidimensional resultante.

Esta metodologia permite uma análise profunda de como o modelo conseguiu discriminar as amostras com base nas características aprendidas durante o treinamento supervisionado, oferecendo *insights* sobre a eficácia do processo de aprendizado e a qualidade das características extraídas.

No proposto trabalho, os processos e experimentos executados podem ser ilustrados na Figura 4.4.

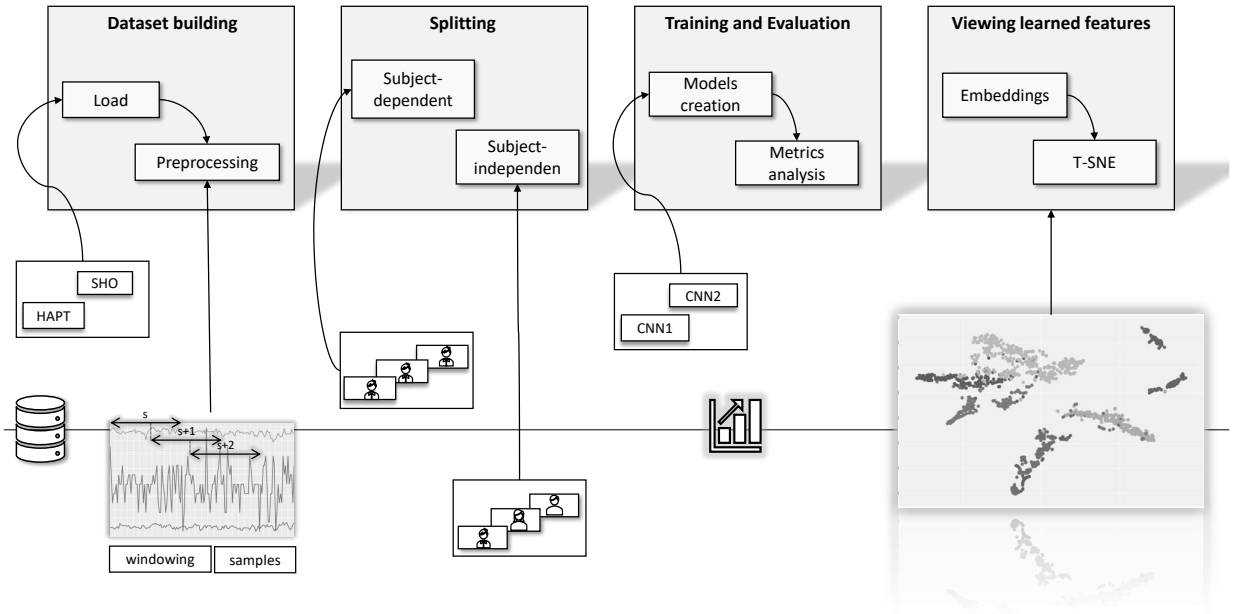


Figura 4.4. Representação esquemática dos procedimentos executados nas bases de dados para obter a visualização utilizando o método t-SNE. Adaptado de (Aquino *et al.*, 2023).

Nos experimentos realizados, para avaliar a qualidade das características extraídas pelos modelos no decorrer do treinamento supervisionado, iniciamos examinando a capacidade do modelo de diferenciar as amostras do conjunto de dados em que foi originalmente treinado. Posteriormente, com o objetivo de investigar a aplicabilidade e eficiência do aprendizado em contextos não familiares, aplicamos a técnica de t-SNE em amostras provenientes de um conjunto de dados distinto daquele usado no treinamento da rede. Esse procedimento nos permite explorar as limitações das características aprendidas pelo modelo inteligente.

4.6 *Framework* proposto

Conforme discutido na Seção 2.5.1, o protocolo padrão de reconhecimento de atividades (ARP) é composto por 5 etapas: aquisição, preprocessamento, segmentação, extração de características, classificação e avaliação. Com essas etapas é possível desenvolver um modelo de inteligência artificial capaz de realizar o reconhecimento de atividades humanas. No entanto, sem a utilização de XAI, os melhores modelos são escolhidos apenas baseado em resultados numéricos de métricas de avaliação. Além disso, não há formas de explicar a tomada de decisão do modelo incorporadas a esse protocolo.

Embora o ARP, seja amplamente utilizado por muitos trabalhos, acreditamos que ainda há margens para melhoria, introduzindo uma nova etapa de explicabilidade, a qual poderia ser aplicada após o treinamento e a avaliação numérica dos modelos profundos. Essa etapa ajudaria na identificação de viés, compreensão do modelo gerado, entre outras aplicações. Um diagrama mostrando o ARP com a etapa adicional proposta é ilustrado na Figura 4.5, e a seguir será discutido mais o que seria essa etapa de explicabilidade.

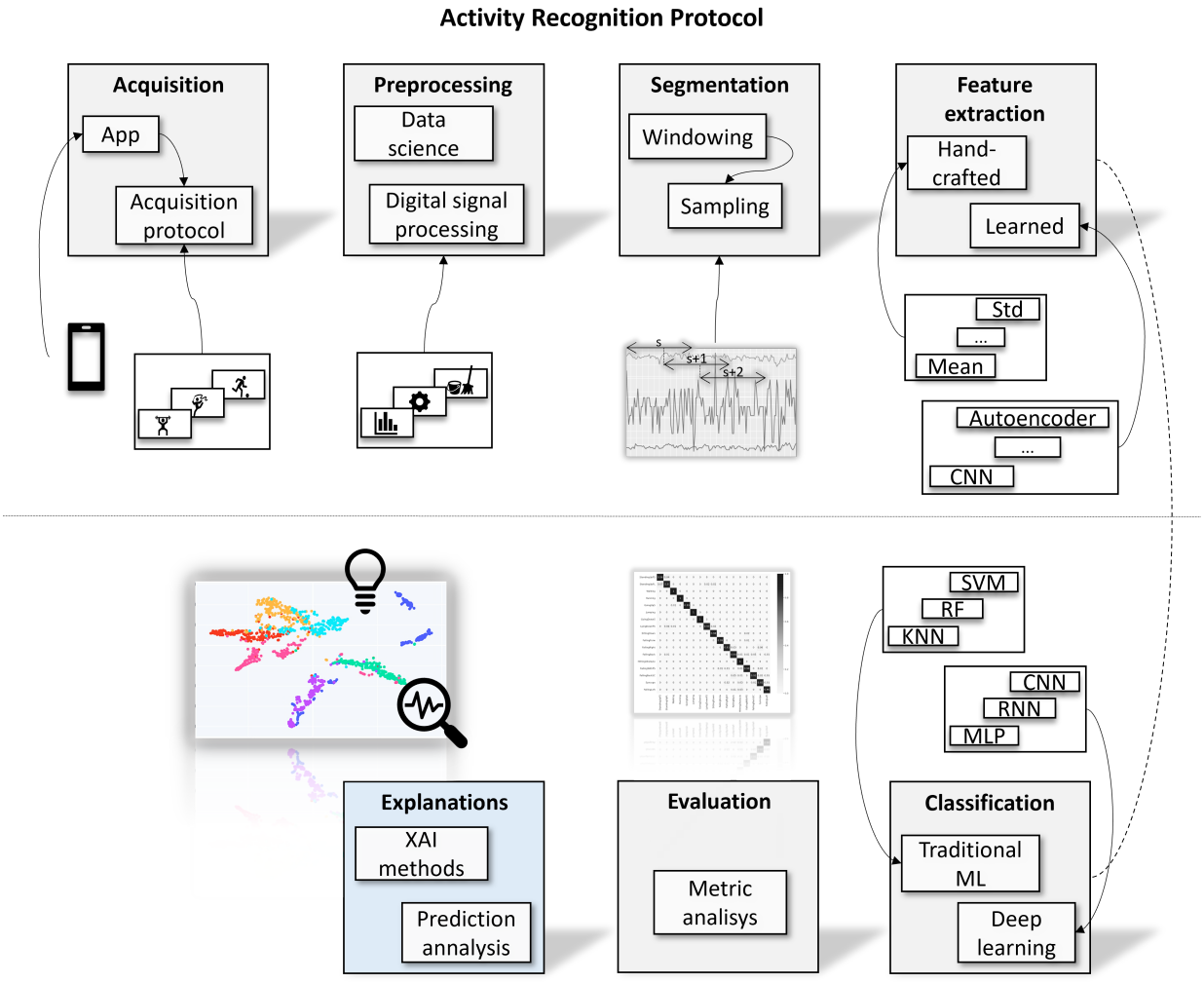


Figura 4.5. O Protocolo Proposto de Reconhecimento de atividades com explicações inclui uma etapa adicional para explicabilidade, permitindo análises além da avaliação métrica. A etapa de classificação utiliza vários algoritmos de aprendizado de máquina, incluindo Maquinas de Vetores de Suporte (SVM), Florestas Aleatórias (RF), K Vizinhos mais Próximos (KNN), Rede Neural Recorrente (RNN) e Perceptron Multicamadas (MLP). Adaptado de (Aquino *et al.*, 2023).

4.6.1 Explicabilidade no protocolo de HAR

Para compreender melhor o processo de tomada de decisão de um modelo, podemos utilizar métodos de Explicabilidade em Inteligência Artificial (XAI). Confiar apenas em resultados numéricos não é a forma mais eficaz de determinar o melhor modelo (Selvaraju *et al.*, 2019). Dois modelos com resultados semelhantes podem ter aprendido características completamente diferentes e se comportarem de maneiras muito distintas. Além disso, um modelo com alto desempenho ainda pode ter aprendido algum viés do conjunto de dados (Aquino *et al.*, 2022, Selvaraju *et al.*, 2019). Portanto, é importante usar técnicas de XAI para obter *insights* sobre como o modelo toma decisões e garantir que as previsões do modelo não sejam baseadas em características enviesadas ou irrelevantes.

Existem duas abordagens distintas em XAI: abordagens transparentes e pós-hoc (Gohel *et al.*, 2021). As abordagens transparentes se referem a modelos cuja arquitetura interna e processo de tomada de decisão são claros e facilmente compreendidos. Exemplos de modelos transparentes incluem o modelo bayesiano, árvores de decisão, regressão linear e sistemas de inferência *fuzzy*. As abordagens transparentes são benéficas quando as correlações internas entre as características não são particularmente complexas nem lineares. Abordagens pós-hoc de explicabilidade podem revelar o funcionamento interno e a lógica de decisão de um modelo de IA treinado, fornecendo pontuações de importância das características, conjuntos de regras, mapas de calor ou linguagem simples para auxiliar os usuários a entender as informações mais relevantes e possível viés (Gohel *et al.*, 2021, Samek *et al.*, 2017). Nesses casos, a técnica pós-hoc pode ser uma ferramenta útil para explicar o que o modelo aprendeu, especialmente quando a relação entre os dados e as características não é simplesmente direta (Gohel *et al.*, 2021).

Dentro dos métodos pós-hoc, temos duas abordagens distintas: agnóstica ao modelo e específica do modelo (Gohel *et al.*, 2021). As abordagens específicas do modelo fornecem limitações de explicabilidade em relação ao algoritmo de aprendizado e à estrutura interna de um determinado modelo de aprendizado profundo. Em contraste, as abordagens agnósticas ao modelo analisam as entradas e as previsões do modelo em pares para compreender os mecanismos de aprendizado e gerar explicações.

Abordagens agnósticas ao modelo, como *Local Interpretable Model-agnostic Explanations* (LIME) e *SHapley Additive Explanations* (SHAP), podem ser usadas com características feitas manualmente e aprendizado de máquina clássico para determinar a importância das características de entrada (Datta *et al.*, 2016, Lipovetsky & Conklin, 2001, Ribeiro *et al.*, 2016).

As abordagens específicas do modelo são usadas em XAI para explicar as decisões tomadas por um modelo de aprendizado de máquina específico. Essas abordagens são adaptadas à arquitetura do modelo, parâmetros e processo de treinamento, e frequentemente dependem do conhecimento dos mecanismos internos do modelo. Exemplos de abordagens específicas do modelo incluem indução de árvore de decisão, extração de regras, métodos baseados em gradientes e propagação de relevância em camadas (Gilpin *et al.*, 2018, Lu *et al.*, 2021, Selvaraju *et al.*, 2019, Zhang *et al.*, 2018).

Essas abordagens fornecem informações valiosas sobre o processo de tomada de decisão do modelo e podem ajudar a melhorar seu desempenho e reduzir viés.

5

Resultados

Este capítulo apresenta os resultados numéricos do trabalho com as seções: (i) Experimentos com a base de dados UniMiB e (ii) Experimentos com as bases de dados SHO HAPT.

5.1 Experimentos com a base de dados UniMiB

5.1.1 Reconhecimento Humano de Atividades

As arquiteturas CNN1 e CNN2 foram treinadas utilizando as estratégias SD e SI. Consideramos as métricas de média macro e ponderada para comparar de forma mais eficaz nossos resultados com outros trabalhos. A diferença entre as métricas é que a primeira leva em consideração apenas o resultado alcançado pela métrica macro em cada classe, enquanto a segunda considera o resultado da métrica em cada classe ponderado pelo número de amostras avaliadas. A média macro lida melhor com um sistema desequilibrado, pois não diferencia classes com volumes maiores de dados. Em contrapartida, a média ponderada fornece uma visão mais geral do sistema, desde que a distribuição das classes no conjunto de dados seja igual à distribuição desses dados no mundo real.

A Tabela 5.1 e a Tabela 5.2 detalham o desempenho de cada rede, considerando as estratégias SD e SI, respectivamente.

Tabela 5.1. Desempenho de classificação das redes CNN1 e CNN2 em HAR, com a estratégia SD e divisão de dados 70/30.

Modelo	**Média macro**			**Média ponderada**			**Acurácia**
	Precisão	**Sensibilidade**	**F1-Score**	**Precisão**	**Sensibilidade**	**F1-Score**	
CNN1	0,9634	0,9624	0,9626	0,9781	0,9779	0,9779	0,9779
CNN2	0,9612	0,9616	0,9610	0,9771	0,9768	0,9768	0,9768

Tabela 5.2. Desempenho de classificação das redes CNN1 e CNN2 em HAR, com a estratégia SI utilizando 9 indivíduos na validação.

Modelo	Média macro			Média ponderada			Acurácia
	Precisão	Sensibilidade	F1-Score	Precisão	Sensibilidade	F1-Score	
CNN1	0,6513	0,6631	0,6530	0,7619	0,7553	0,7565	0,7553
CNN2	0,6361	0,6361	0,6328	0,7321	0,7177	0,7207	0,7177

Os resultados para a estratégia de validação SD foram significativamente superiores aos obtidos com a estratégia SI para ambas as arquiteturas treinadas, considerando todas as métricas. Esse resultado já era esperado, pois estudos anteriores indicam que o método de validação SI é mais desafiador e se aproxima mais do resultado real que uma rede terá, uma vez que a rede não terá conhecimento prévio sobre como os indivíduos realizam a atividade (Ferrari *et al.*, 2021).

Nos dois cenários avaliados, a CNN1 foi levemente superior à CNN2. No entanto, com a estratégia SD, os resultados entre as redes foram próximos, apresentando uma diferença entre os valores do macro F1-score obtidos de $0,2\%$. Na estratégia SI, já é possível observar uma diferença de desempenho mais significativa de aproximadamente 2% para a métrica macro F1-score e 4% para a métrica de precisão.

Embora o foco do nosso artigo não seja obter a melhor arquitetura possível nas áreas de HAR e BUI, os resultados obtidos são tão bons quanto, ou até melhores que, os artigos apresentados na Tabela 3.1. Apesar dos métodos de validação e subconjuntos serem diferentes, vale a pena destacar o excelente desempenho do modelo. Observando a métrica de precisão, nosso trabalho alcançou a mesma precisão que o melhor trabalho com a estratégia SD, $0,978$, usando a rede CNN1. A diferença é que este resultado foi alcançado com validação cruzada. Considerando que a validação cruzada superestima o desempenho da rede, o resultado obtido com o método *hold-out* é significativo (Bragança *et al.*, 2022). Uma análise completa dos trabalhos anteriores pode ser feita observando a Tabela 3.1, então incluímos todas as nossas métricas de desempenho obtidas com a CNN1.

O resultado para a estratégia SI foi de $0,755$ em precisão, que também se aproxima da base de comparação. Mais uma vez, os resultados alcançados neste artigo são mais expressivos, visto que o melhor resultado foi obtido com o método LOSO, que possui apenas um indivíduo na validação. O nosso método possui nove indivíduos no conjunto de validação.

As métricas de desempenho mostram que a melhor estratégia de validação foi SD e a melhor rede foi CNN1. A seguir, utilizaremos outros métodos de avaliação e validação para contrastar essas arquiteturas e escolher tanto o melhor modelo quanto a melhor abordagem de validação. A seguir, analisaremos as diagonais das matrizes de

confusão mostradas nas Figuras 5.1 e 5.2. A diagonal é nossa métrica de sensibilidade, previamente definida na Tabela 2.1. Esta métrica também é chamada de Taxa de Verdadeiro Positivo (TPR). Como mostrado na Figura 5.1, para a CNN1 validada com SD, entre as ADLs, a classe com menor desempenho foi *StandingUpFL* com 90% de TPR, seguida por *LyingDownFS* com 93% de TPR. A confusão sobre *StandingUpFL* foi bastante compreensível, pois 7% das amostras previstas erroneamente foram confundidas com *StandingUpFS*, que são ADLs de fato similares. No entanto, entre as confusões de *LyingDownFS*, não parece haver um sentido lógico, pois a maior parte da confusão, aproximadamente 4%, ocorreu com a classe *StandingUpFS*. Observando o desempenho em queda das classes, a confusão parece ocorrer entre os diferentes tipos de queda. A classe com menor desempenho foi a classe *FallingRight* com 91% de TPR, frequentemente confundida com a classe *Syncope*.

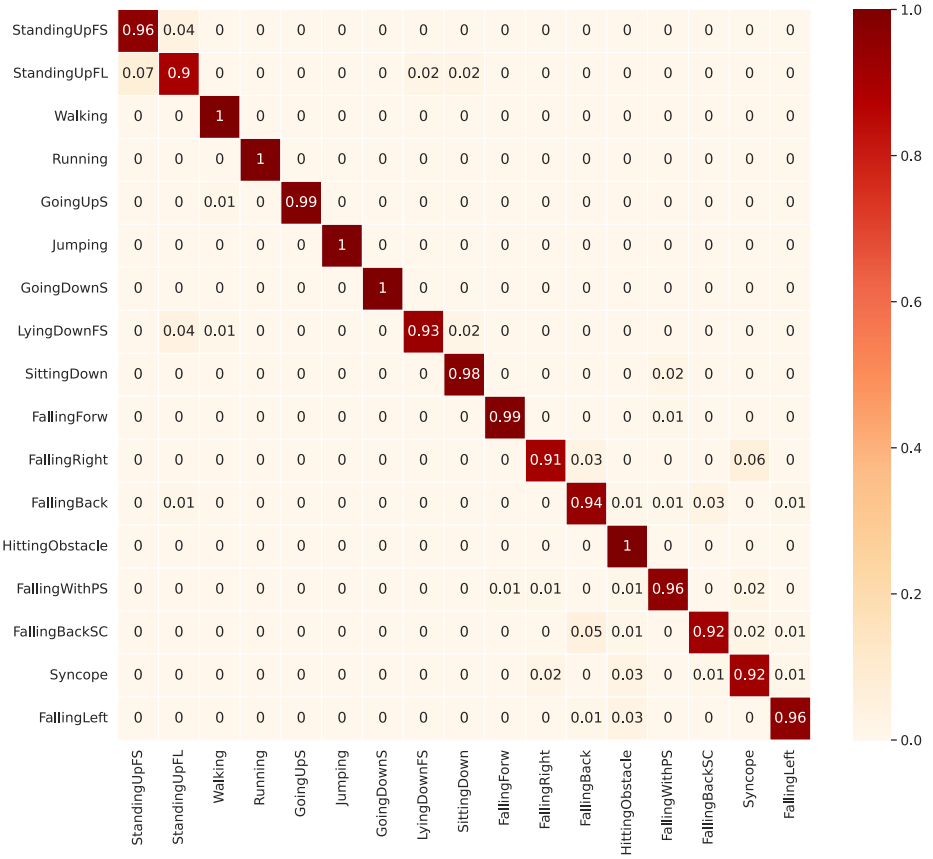


Figura 5.1. Matriz de confusão da rede CNN1 com o método de validação sujeito dependente (SD) utilizando proporção 70/30

A Figura 5.2 revela que, entre as ADLs, o desempenho mais baixo para a CNN1 validada com SI foi observado na classe *SittingDown*, com uma taxa de verdadeiro positivo (TPR) de 44%, seguida por *LyingDownFS* com 66% de TPR. A confusão na classe

SittingDown sugere uma possível deficiência no aprendizado da rede, visto que 37% das amostras foram erroneamente associadas à classe GoingDownS, atividades estas que são completamente distintas. A maior parte da confusão na classe *LyingDownFS*, correspondendo a 16% do total, ocorreu com a classe *SittingDown*, consolidando a percepção de que os atributos aprendidos pela classe *SittingDown* podem não ter sido adequados. No Capítulo 6, destacaremos que as classes *SittingDown* e *LyingDownFS* apresentam certos vieses que podem comprometer o aprendizado da rede. No que tange às classes relacionadas a quedas, identificou-se um padrão semelhante ao observado em SD. Grande parte da confusão ocorre entre os diferentes tipos de queda, mas também com classes de ADLs, como é o caso de *FallingBackSC*, onde 6% das amostras foram confundidas com *LyingDownFS*.

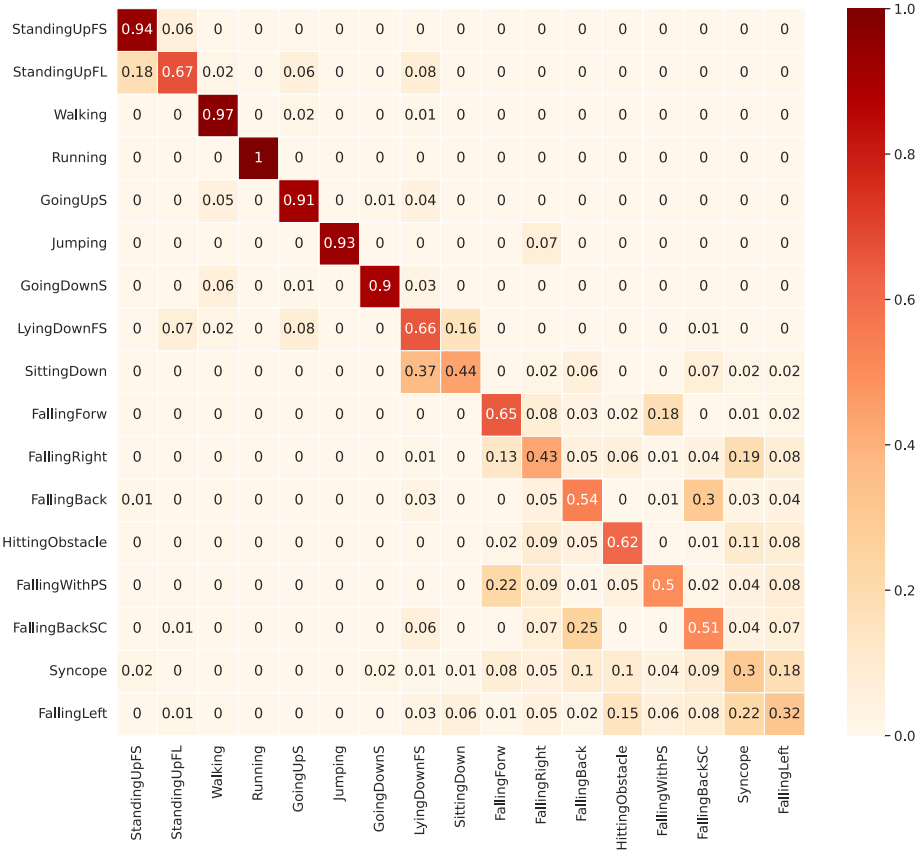


Figura 5.2. Matriz de confusão da rede CNN1 com o método de validação sujeito independente (SI) utilizando 9 indivíduos na validação

De modo geral, no SI, para algumas classes, os atributos aprendidos por determinados sujeitos foram suficientes para classificar a atividade realizada por outros sujeitos, como é o caso da classe *Running*. Em SD, dado que a rede processa amostras de todos os indivíduos, a classificação torna-se mais simplificada, o que se reflete em

desempenhos superiores. No entanto, não é possível determinar se esses atributos são suficientemente relevantes para obter um bom desempenho em sujeitos que não fazem parte da base de dados, uma vez que não dispomos de uma parcela independente de dados com a mesma distribuição para validar os resultados.

5.1.2 Identificação biométrica de usuário

Para demonstrar que as amostras de dados HAR contêm uma vasta quantidade de informações pessoais do usuário, treinamos e avaliamos a arquitetura CNN1 para realizar a identificação do sujeito com base na atividade realizada pelo indivíduo. Este sistema, quando em produção, deve ser precedido por um classificador que detecta a atividade executada pelo usuário e, só então, reconhece o sujeito, uma vez que cada IA de identificação foi treinada e validada após filtrar a base de dados por atividade. Utilizando a divisão de dados SD apresentada na Figura 2.4, modificamos apenas o número de neurônios na última camada da CNN1, Softmax, para o número de sujeitos que executou cada atividade, conforme a coluna N° de sujeitos na Tabela 4.1. Foram obtidas dezessete redes, uma para cada atividade. A Figura 5.3 apresenta o desempenho de cada IA desenvolvida com base na métrica macro F1-score.

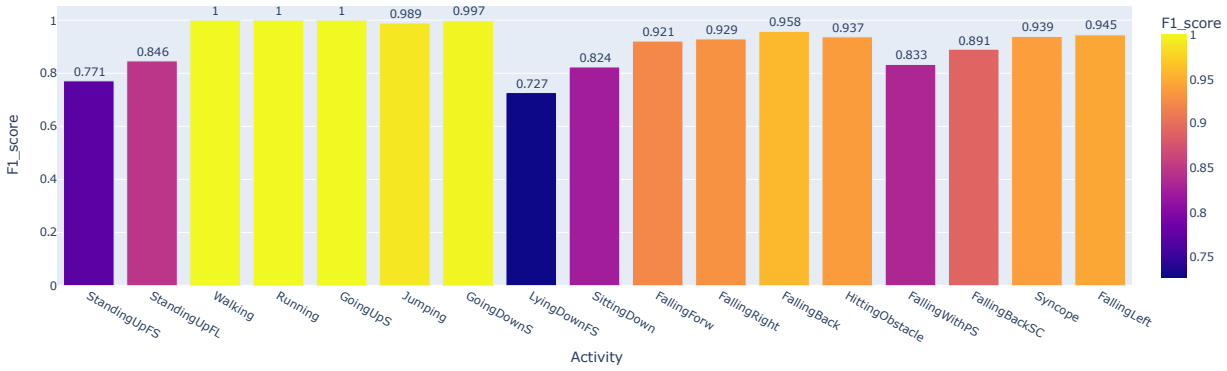


Figura 5.3. Desempenho geral para BUI para cada atividade física ou queda considerando a métrica F1-score macro. Adaptado de (Aquino *et al.*, 2022).

Em algumas atividades, a identidade do usuário é mais marcante do que em outras. Por exemplo, nas amostras de *Walking*, é possível reconhecer cada um dos indivíduos com uma macro F1-score de 100%. Já para o subconjunto *LyingDownFS*, este valor cai para 73%. Entre as atividades físicas diárias mais comuns que alcançaram os melhores desempenhos estão Caminhada (*Walking*), Corrida (*Running*), Subindo Escadas (*GoingUpS*) e Descendo Escadas (*GoingDownS*).

Mais adiante, na seção de discussão, associaremos o desempenho no reconhecimento da atividade ao desempenho na identificação do sujeito.

Em resumo, observa-se que é possível reconhecer bem os indivíduos, com uma média superior a 91% de todas as macro F1-scores obtidas para cada subconjunto. Na próxima seção, demonstraremos que as classes *StandingUpFS*, *StandingUpFL*, *LyingdownFS* e *SittingDown* são atividades que sofrem de problemas de viés na base de dados. Estas foram precisamente os subconjuntos de menor desempenho de nossa IA para identificação de usuário, indicando que o desempenho desta rede para identificação seria ainda mais elevado se desconsiderássemos o desempenho para estas atividades.

5.2 Experimentos com as bases de dados SHO e HAPT

5.2.1 SHO

Neste estudo, avaliamos e treinamos duas arquiteturas convolutivas, CNN1 e CNN2, empregando as estratégias SD e SI. Para possibilitar uma comparação mais precisa com outros estudos relevantes, utilizamos métricas de média macro e média ponderada. Os resultados detalhados do desempenho de cada rede, considerando as estratégias SD e SI, são apresentados nas Tabelas 5.3 e 5.4.

Tabela 5.3. Desempenho da classificação das CNN1 e CNN2 no HAR, com a estratégia SD com divisão de treino e validação 70/30 no bando de dados SHO.

Modelo	Média macro			Média ponderada			Acurácia
	Precisão	Sensibilidade	F1-Score	Precisão	Sensibilidade	F1-Score	
CNN2	0,9976	0,9976	0,9976	0,9976	0,9976	0,9976	0,9976
CNN1	0,9966	0,9969	0,9967	0,9968	0,9968	0,9968	0,9968

Tabela 5.4. Desempenho da classificação das CNN1 e CNN2 no HAR, com a estratégia SI com 3 indivíduos na validação para o bando de dados SHO.

Modelo	Média macro			Média ponderada			Acurácia
	Precisão	Sensibilidade	F1-Score	Precisão	Sensibilidade	F1-Score	
CNN2	0,9816	0,9814	0,9814	0,9816	0,9814	0,9813	0,9814
CNN1	0,9786	0,9782	0,9781	0,9786	0,9782	0,9781	0,9782

Os resultados obtidos com a estratégia de validação SD foram marginalmente superiores aos adquiridos com a estratégia SI para ambas as arquiteturas treinadas, em todas as métricas. Este resultado era esperado com base em pesquisas anteriores, que sugerem que o método de validação SI é mais rigoroso e oferece uma representação mais precisa do desempenho da rede em cenários do mundo real, pois a rede não é fornecida com conhecimento prévio sobre como os indivíduos realizam uma atividade.

A arquitetura CNN2 superou a CNN1 usando a estratégia SD, enquanto, com a estratégia SI, a CNN1 teve um desempenho melhor. O desempenho de ambas as arquiteturas foi muito próximo, com uma diferença de apenas 0,02% na estratégia SD e 0,3% na estratégia SI.

A abordagem SI foi determinada como mais exigente e mais alinhada com a forma como o modelo seria avaliado em cenários do mundo real. Consequentemente, maior ênfase foi colocada nos resultados do SI. Ao examinar a matriz de confusão mostrada na Figura 5.4, obtivemos *insights* sobre os desafios encontrados pelo modelo CNN1 ao utilizar a abordagem SI.

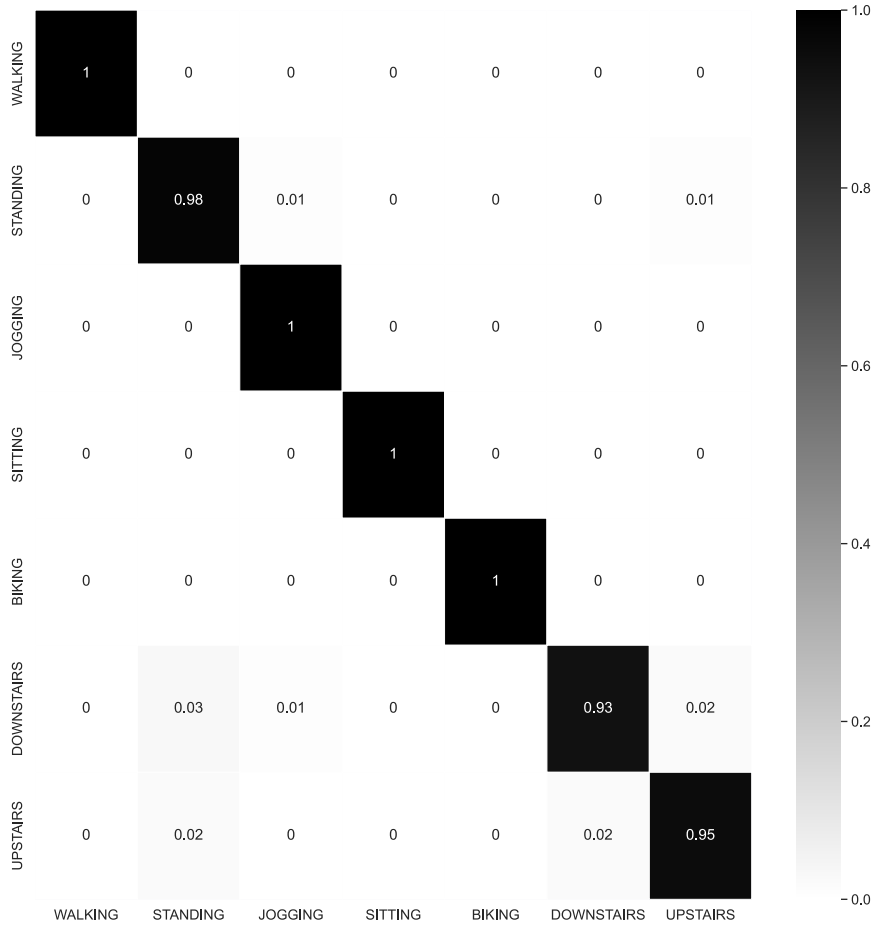


Figura 5.4. Matriz de confusão da CNN1 com independência de sujeito, com 3 sujeitos na validação para todas as 8 classes no conjunto de dados SHO.

5.2.2 HAPT

Uma vez que esta base de dados é altamente desbalanceado, os resultados eram esperados ser menos notáveis, dado que havia 12 classes e apenas algumas amostras das

atividades de transição postural. Os resultados são exibidos nas Tabelas 5.5 e 5.6.

Tabela 5.5. Desempenho da classificação das CNN1 e CNN2 no HAR, com a estratégia SD e divisão de 70/30.

Modelo	Média macro			Média ponderada			Acurácia
	Precisão	Sensibilidade	F1-Score	Precisão	Sensibilidade	F1-Score	
CNN2	0,8724	0,8817	0,8742	0,9515	0,9509	0,9508	0,9509
CNN1	0,8749	0,8706	0,8723	0,9482	0,9481	0,9406	0,9481

Tabela 5.6. Desempenho da classificação das CNN1 e CNN2 no HAR, com a estratégia SI e 9 sujeitos na validação.

Modelo	Média macro			Média ponderada			Acurácia
	Precisão	Sensibilidade	F1-Score	Precisão	Sensibilidade	F1-Score	
CNN2	0,8734	0,8632	0,8638	0,9289	0,9275	0,9274	0,9275
CNN1	0,8624	0,8603	0,8546	0,9259	0,9239	0,9238	0,9239

Os resultados obtidos entre as duas abordagens foram comparáveis. Considerando o F1-Score médio, houve uma diferença de menos de 1% entre a melhor arquitetura usando as abordagens SI e SD. Essa diferença foi inferior a 3% em termos da métrica de acurácia. As métricas ponderadas alcançaram os melhores resultados em ambas as abordagens, como esperado, devido à natureza desbalanceada do conjunto de dados.

A Figura 5.5 exibe uma matriz de confusão para a melhor arquitetura utilizando a abordagem SI.

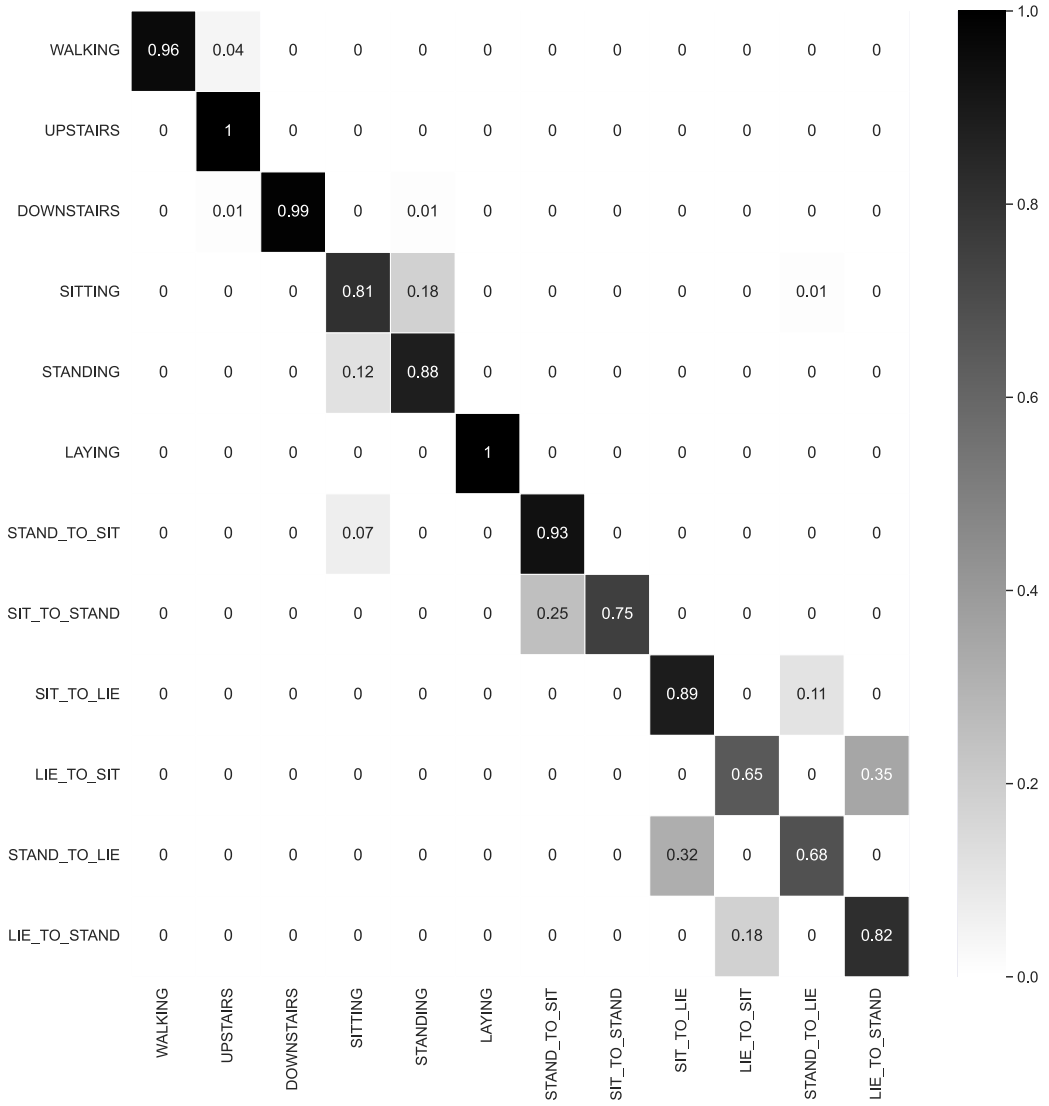


Figura 5.5. Matriz de confusão da CNN1 para a abordagem SI, com 9 sujeitos no conjunto de validação, usando todas as 12 classes no conjunto de dados HAPT.

6

Discussões

Este capítulo apresenta as discussões a partir da análise dos resultados utilizando técnicas de XAI tanto para BUI quanto para HAR com as seções: (i) Explicando a correlação entre HAR e BUI, (ii) Explicando modelos de CNN aplicados a HAR com grad-CAM, (iii) Utilizando grad-CAM para identificar viés em um dataset e (iv) Análises XAI com a aplicação do método t-SNE.

6.1 Explicando a correlação entre HAR e BUI

A Tabela 4.1 revela que nem todos os indivíduos executaram todas as atividades no conjunto de dados UniMiB-SHAR. Ademais, conforme demonstrado na Tabela 3.1, diversos trabalhos adotam a abordagem SI com uma estratégia de validação *leave-one-subject-out*. Esta estratégia de validação não é apropriada para este cenário, dado que a maioria dos modelos de aprendizado de máquina requer dados Independentes e Identicamente Distribuídos (IID) (Wong, 2015). Dado que nem todos os indivíduos realizaram todas as atividades neste conjunto de dados, a utilização desta estratégia não atende aos critérios para se ter dados IID.

Além disso, muitos trabalhos afirmam ter superado seu *baseline* de desempenho e serem o estado da arte atual do conjunto de dados UniMiB-SHAR (Juefei-Xu *et al.*, 2012, Mukherjee *et al.*, 2020, Ronao & Cho, 2016, Tang *et al.*, 2020, Teng *et al.*, 2021). Contudo, algumas pesquisas não empregam a mesma estratégia de validação que seu *baseline*, tornando os resultados incomparáveis. Alguns trabalhos utilizam a mesma estratégia de validação com diferentes subconjuntos de dados (Lv *et al.*, 2020, Tang *et al.*, 2020). Por exemplo, a maioria das divisões de dados é feita de forma aleatória, obedecendo uma proporção de subdivisão, mas cada divisão deve ser baseada na mesma semente aleatória para assegurar a reprodutibilidade dos resultados. No entanto, alguns

autores desconsideram este fato, obtendo subconjuntos não equivalentes, o que implica que, em alguns casos, podemos ter amostras similares nos conjuntos de treinamento e validação, elevando o desempenho. Em contraste, em outros casos, pode haver menos amostras similares nos subconjuntos de treinamento e validação, levantando o problema da generalização da rede e, provavelmente, resultando em desempenho inferior.

No Capítulo 5, demonstrou-se que as redes reconhecem todas as 17 atividades de forma eficaz, porém há uma diferença superior a 31% entre as duas estratégias de validação no que tange à métrica de macro F1-score. Também se observa um alto desempenho na identificação de usuários por atividade. A Figura 6.1 apresenta, para média dos resultados de todas as atividades, valores de reconhecimento de HAR com SD, HAR com SI e BUI.

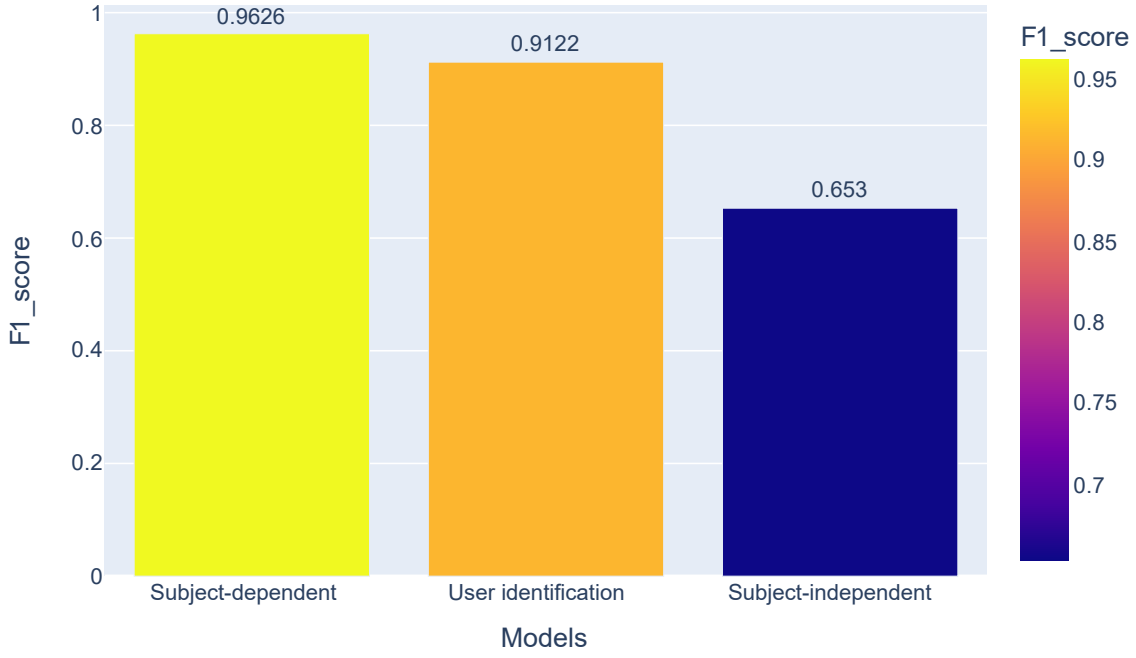


Figura 6.1. Comparação entre os melhores resultados obtidos para identificação de HAR com SD, HAR com SI e BUI. Adaptado de (Aquino *et al.*, 2022).

A Figura 6.2 ilustra uma correlação entre o desempenho de reconhecimento de Atividades da Vida Diária (ADLs, do inglês *Activities of Daily Living*) com a estratégia de Divisão de Sujeito Dependente (SD) versus o desempenho de identificação do usuário (BUI) para essas ADLs. O coeficiente de Pearson e o valor-p para o teste de não correlação são, respectivamente, 0,775 e 0,014. O coeficiente de Pearson foi descrito em mais detalhes na seção 2.2.1. A hipótese nula (H_0) deste experimento assume que não existe correlação entre as duas variáveis analisadas. Considerando um nível de significância (α) de 0,05, o baixo valor-p (0,014) permite rejeitar a hipótese nula,

indicando que a correlação observada é estatisticamente significativa. Isso sugere que o alto desempenho obtido com a estratégia de validação SD pode estar relacionado ao modelo aprender não somente a atividade física, mas também a assinatura do indivíduo. Esse resultado será mais discutido na Seção 6.2.

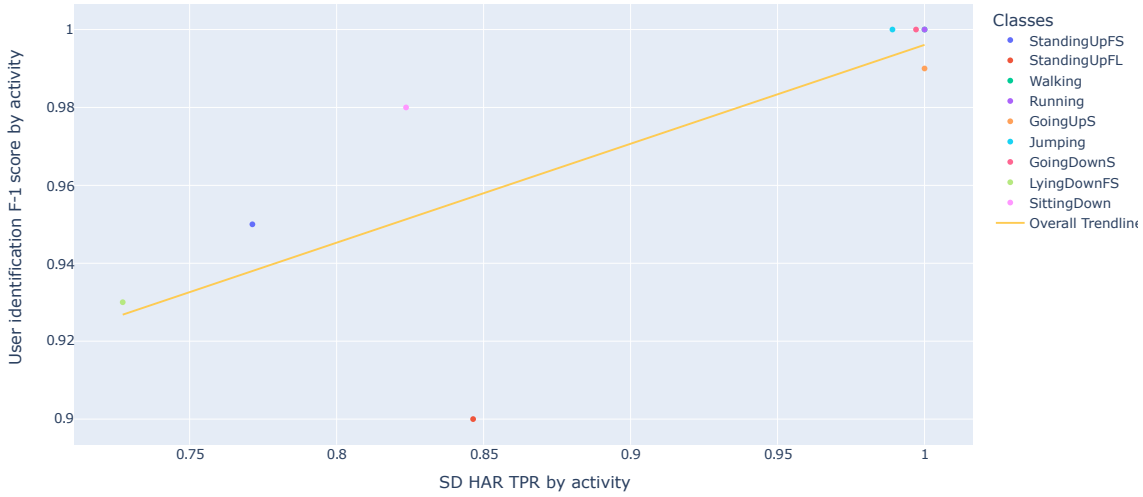


Figura 6.2. Correlação entre os resultados de HAR SD com BUI considerando somente as atividades físicas de ADL do UniMiB-SHAR. Adaptado de (Aquino *et al.*, 2022).

Como ilustrado na Figura 6.3, considerando tanto as ADLs quanto as Quedas, o coeficiente de Pearson e o valor-p obtidos foram 0,504 e 0,039, respectivamente. Quando levamos em conta as quedas, a correlação é mais fraca. No entanto, é importante destacar que as quedas geralmente não são utilizadas para reconhecer o usuário da mesma forma que as ADLs.

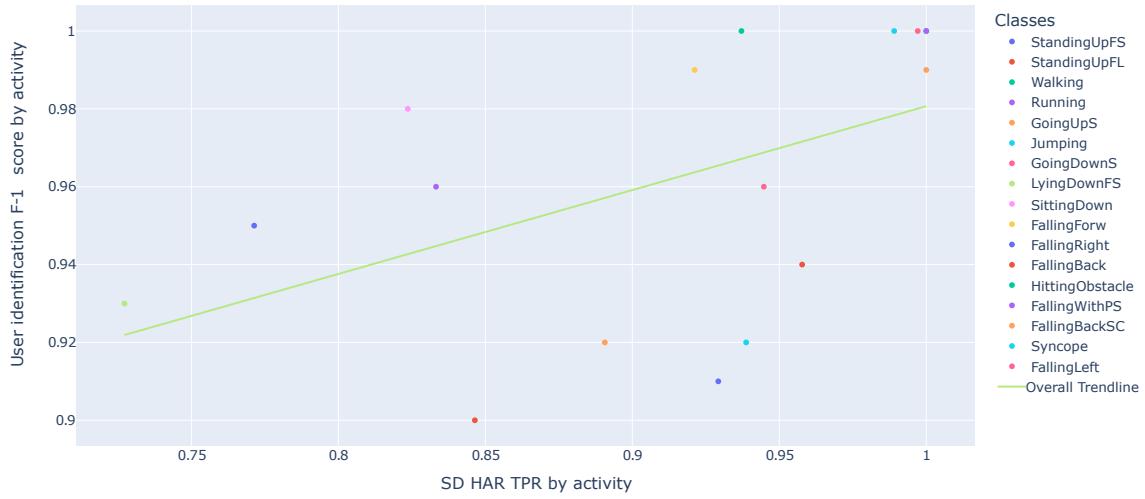


Figura 6.3. Correlação entre os resultados de HAR SD com BUI considerando todas as atividades físicas de ADL e queda do UniMiB-SHAR. Adaptado de (Aquino *et al.*, 2022).

De maneira geral, em cada atividade analisada, o desempenho do classificador treinado com a estratégia SD pode não ser um indicativo confiável de quão eficazmente esse classificador reconhecerá as atividades em cenários da vida real.

6.2 Explicando modelos de CNN aplicados a HAR com grad-CAM

A interpretação das predições geradas pelos modelos é fundamental para avaliar sua capacidade de generalização e entender suas limitações. A técnica grad-CAM oferece um meio de realizar esta análise, fornecendo informações visuais sobre as regiões do sinal de entrada que mais influenciam a tomada de decisão do modelo. A Figura 6.4 ilustra o que a arquitetura CNN1 aprendeu e considerou para a previsão da classe *Walking*.

Figura 6.4. Segmentos de maior importância para tomada de decisão na arquitetura CNN1, para HAR com SI, HAR com SD e identificação do usuário, considerando amostras de Caminhada com diferentes indivíduos. Quanto mais escura a área, mais crítica ela é para a tomada de decisão. Para obter esses mapas de calor, um limiar de $0,7$ foi considerado no resultado do grad-CAM a fim de tornar mais evidentes as regiões de maior importância. Adaptado de (Aquino *et al.*, 2022).

Existem padrões-chave que auxiliam na identificação de cada atividade. Uma rede pode aprender esse evento-chave ou outros padrões ocultos para identificar uma atividade. No entanto, quanto maior a região que a arquitetura considera para a tomada de decisão, menor é a probabilidade de ter aprendido apenas esses padrões-chave. A análise preliminar da Figura 6.4, permite afirmar que as regiões de BUI são sempre maiores que nas abordagens SI e SD, tendo SD uma região levemente mais vasta, permitindo teorizar que para as amostras analisadas de SD, uma parcela das identidades dos indivíduos tenha sido aprendida durante o treinamento. O fato de BUI precisar de uma região maior do sinal faz sentido, uma vez que ele precisa obter uma assinatura geral do indivíduo.

Algumas atividades possuem múltiplos eventos-chave em uma única amostra. Por exemplo, na atividade de Corrida, onde eventos-chave podem estar associados a passos, no período de 3 segundos de uma amostra, o número de eventos-chave é maior do que na atividade de Caminhada. Na Figura 6.5 é possível notar uma diferença grande de tamanho da região de importância de BUI com as demais abordagens. No entanto, para duas das amostras a diferença da região de importância entre SD e SI é mais sutil, quando comparada a Figura 6.4.

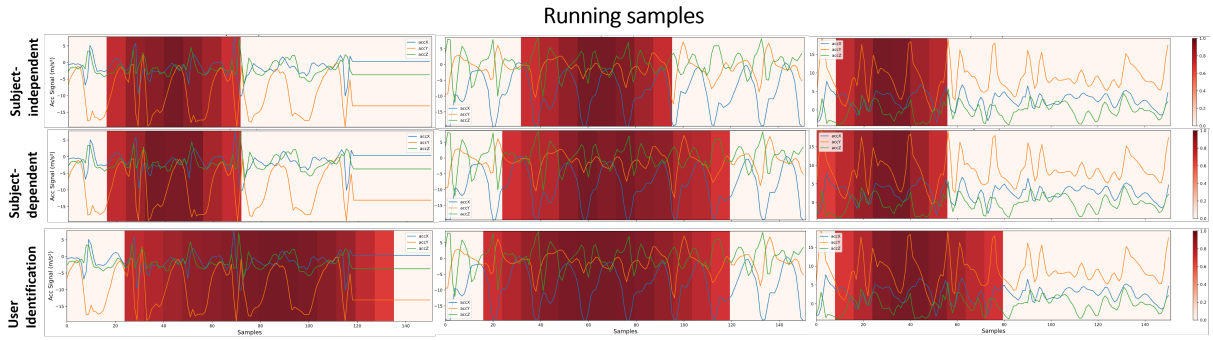


Figura 6.5. Segmentos de maior importância para tomada de decisão na arquitetura CNN1, para HAR com SI, HAR com SD e identificação do usuário, considerando amostras de Corrida com diferentes indivíduos. Quanto mais escura a área, mais crítica ela é para a tomada de decisão. Para obter esses mapas de calor, um limiar de $0,7$ foi considerado no resultado do grad-CAM a fim de tornar mais evidentes as regiões de maior importância. Adaptado de (Aquino *et al.*, 2022).

Preliminarmente podemos observar um padrão de região de importância maior para BUI com relação as demais abordagens. Além disso, é possível notar outro padrão através da análise das visualizações com grad-CAM. Como demonstrado na Figura 4.2, a arquitetura CNN2, quando comparada à arquitetura CNN1, possui uma resolução mais elevada (número de neurônios) na última camada convolucional. Enquanto a CNN1 tem uma resolução de 19×1, a CNN2 tem 38×1. A Figura 6.6 demonstra que esta maior resolução permite que a rede aprenda eventos-chave com maior precisão. Contrastamos o aprendizado entre as abordagens SI e SD através desta figura.

Decidimos remover as visualizações das redes de identificação de usuário, uma vez que nas amostras observadas BUI sempre considera uma região maior do sinal, conforme ilustrado nas Figuras 6.4 e 6.5. Observamos que as estratégias SI e SD aprendem eventos-chave, mas a rede SD também atribui importância secundária ao sinal como um todo. O que a rede está assimilando entre os eventos-chave é a assinatura do usuário. A implicação disto no resultado final é que a rede pode memorizar como os usuários na base de dados realizam atividades, ao invés de aprender os padrões específicos da atividade física. As porções de importância secundária entre os eventos-chave na rede SI estão menos dispersas do que na rede SD.

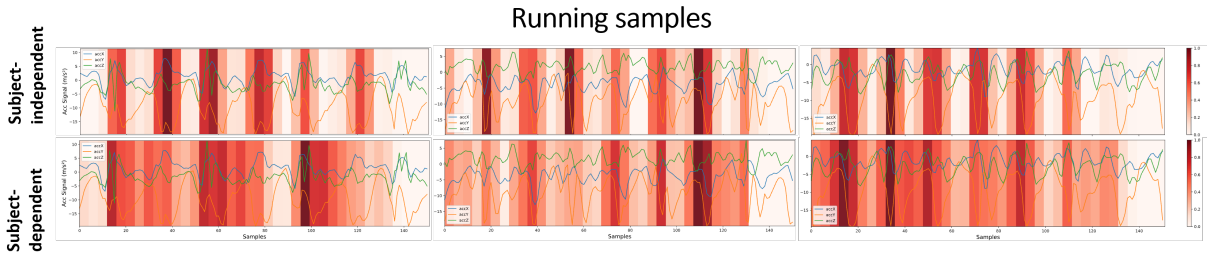


Figura 6.6. Segmentos de maior importância para a tomada de decisão na arquitetura CNN2, para HAR com SI e HAR com SD, considerando amostras da classe Corrida com diferentes indivíduos. Para obter os mapas de calor, não foi considerado um limiar no resultado do grad-CAM a fim de tornar mais transparentes as zonas de importância menores, mas significativas, que a estratégia SD leva em conta. Adaptado de (Aquino *et al.*, 2022).

A seguir, é feita uma análise de mais alto nível, onde é possível contrastar as redes CNN1 x CNN2, bem como as estratégias de validação SD e SI, como ilustrado na Figura 6.7.

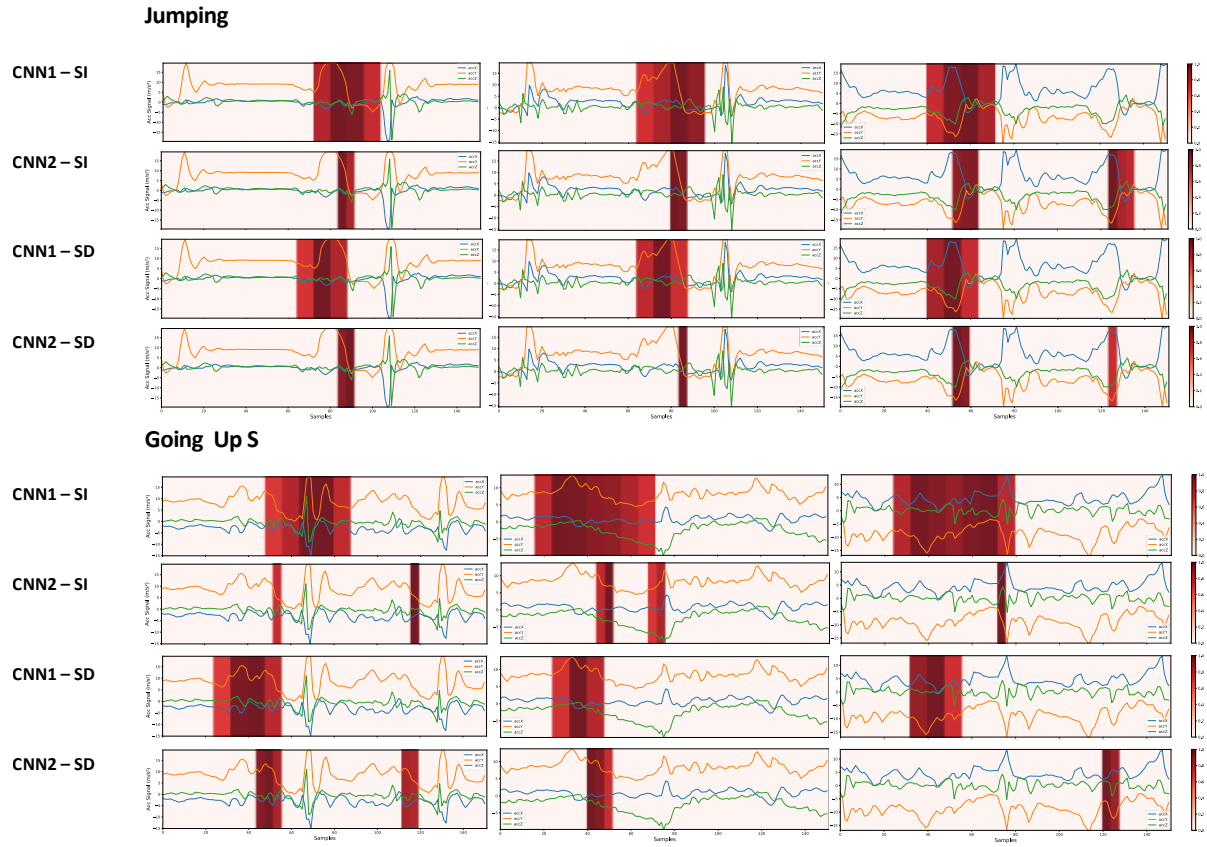


Figura 6.7. Segmentos mais importantes para a tomada de decisão nas redes CNN1 e CNN2 com as estratégias SD e SI, para amostras das atividades Saltar e Subir Escadas. Para obter os mapas de calor, foi considerado um limiar de $0,7$ no resultado do grad-CAM para tornar as regiões críticas mais evidentes. Adaptado de (Aquino *et al.*, 2022).

Comparando-se as redes, observa-se na Figura 6.7 o mesmo padrão observado na Figura 6.6, de que as regiões de importância são menores para a CNN2, o que não necessariamente significa que o modelo baseado na CNN2 é melhor. Por exemplo, a rede CNN1 aprendeu características mais complexas do sinal ao considerar uma região mais extensa para a tomada de decisões. Simultaneamente, a rede CNN2 aprendeu eventos menos complexos que podem ocorrer mais facilmente em amostras fora da base de dados, levando a predições incorretas.

Analisando-se o comportamento para as abordagens de validação para a Figura 6.7, percebe-se que não se pode afirmar que a região de importância é sempre maior para SD comparada a SI. As redes treinadas com a abordagem SI, não sofrem o risco de aprender algo indesejado como a assinatura individual do usuário, uma vez que há indivíduos diferentes na validação. Dessa forma, mesmo não podendo confirmar a hipótese de que o aprendizado por SD pode estar relacionado ao aprendizado de assinatura do usuário, a análise realizada pelo coeficiente de Pearson leva a acreditar que este é o caso. Talvez para confirmar este estudo seja necessário uma base de dados com muito mais dados de diferentes indivíduos disponíveis, sendo assim possível retirar indivíduos para a validação, sem modificar drasticamente a distribuição do conjunto de dados.

Estudos recentes na área XAI têm focado não apenas na interpretabilidade dos modelos, mas também na qualidade e na confiabilidade das explicações visuais fornecidas por esses métodos Adebayo *et al.* (2020), Hedström *et al.* (2023), Samek *et al.* (2017). A aplicação de abordagens sólidas em XAI visa assegurar que as conclusões qualitativas sobre o funcionamento e as decisões dos modelos sejam fundamentadas em análises precisas e transparentes. Para tanto, esses métodos avaliam a qualidade das explicações sob diferentes prismas, tais como a fidelidade, que mensura em que medida as explicações acompanham o comportamento preditivo do modelo, destacando a importância das características mais relevantes para as decisões do modelo. A robustez, por sua vez, avalia a estabilidade das explicações diante de pequenas perturbações nos dados de entrada, pressupondo que a saída do modelo permaneça aproximadamente constante. A localização verifica se as evidências explicativas estão centradas em uma região de interesse, definida, por exemplo, por uma caixa delimitadora ou uma máscara de segmentação. A complexidade captura o quão concisas são as explicações, privilegiando aquelas que utilizam um menor número de características para explicar uma previsão do modelo. Por fim, a randomização testa em que medida as explicações se deterioram conforme os dados ou os parâmetros do modelo são progressivamente randomizados. Essas abordagens multidimensionais de avaliação possibilitam não apenas uma melhor compreensão dos modelos, mas também promovem o desenvolvimento

de métodos XAI mais eficazes e confiáveis, capazes de fornecer *insights* valiosos e facilmente interpretáveis pelos usuários finais Adebayo *et al.* (2020), Hedström *et al.* (2023).

Embora os métodos de avaliação de explicações visuais em XAI ofereçam *insights* valiosos sobre o funcionamento de modelos de aprendizado profundo, sua aplicabilidade direta a HAR a partir de dados unidimensionais, como séries temporais, encontra limitações significativas. No contexto do HAR, os sinais capturados, como os provenientes de acelerômetros, são essencialmente sequências temporais contínuas sem marcações específicas realizadas por especialistas que delimitam exatamente quais segmentos dos dados correspondem a determinadas atividades. Por exemplo, amostras etiquetadas como "caminhada"podem ter uma duração padrão, como 3 segundos, mas não possuem uma região exata dentro do sinal que identifique univocamente a caminhada. Esta natureza dos dados implica que os conceitos de fidelidade, localização, e outras métricas desenvolvidas para avaliar a qualidade das explicações em contextos onde as características importantes são espacialmente ou visualmente localizáveis, podem não ser diretamente aplicáveis ou necessitam de adaptações significativas. Desta forma, não iremos abordar ou aplicar um método para mensurar a qualidade da explicações fornecidas.

De forma geral, um modelo pode apresentar robustez quando avaliado através de suas métricas de desempenho. No entanto, ele pode não ser capaz de generalizar bem em cenários do mundo real. Conforme demonstrado na Figura 6.7, embora as arquiteturas CNN1 e CNN2 treinadas com a estratégia SD aprendam características distintas, suas métricas de desempenho diferiram em apenas $0,2\%$. A rede CNN1 tem a capacidade de aprender características mais complexas do que a rede CNN2. Além disso, concluímos que a rede CNN1 aparenta ser mais robusta, na medida em que não dá importância a eventos-chave tão breves quanto a CNN2, tornando essa rede menos suscetível a ruído. Tal análise não seria possível sem a exploração visual proporcionada pela técnica grad-CAM.

6.3 Utilizando grad-CAM para identificar viés em um conjunto de dados

Os modelos de ML são intrinsecamente orientados por dados, fundamentando-se na premissa de aprender a partir das informações que lhes são fornecidas. Contudo, a integridade e representatividade desses dados são cruciais. Se os dados apresentarem ruídos, inconsistências ou vieses, o modelo inevitavelmente incorporará tais imperfei-

ções.

A análise rigorosa de conjuntos de dados é uma tarefa primordial, especialmente quando se trata de bases de dados volumosas e desbalanceadas. Enquanto algumas inconsistências, como dados faltantes, são relativamente simples de serem identificadas, outras irregularidades, mais sutis, podem ser desafiadoras de serem detectadas e corrigidas. Uma ferramenta promissora nesse contexto é a exploração visual através do grad-CAM, que pode ajudar a identificar vieses não imediatamente aparentes.

No estudo conduzido por Selvaraju *et al.* (2019), o grad-CAM foi empregado para caracterizar um viés inesperado em um modelo de IA. No referido estudo, os autores buscaram desenvolver um modelo para classificar imagens de médicos e enfermeiros, independentemente do gênero do indivíduo retratado. Embora os indicadores de desempenho sugerissem uma boa generalização do modelo no conjunto de dados de treinamento, quando submetido a imagens externas, o comportamento do modelo foi notadamente diferente. No conjunto de dados de treino a maior parte dos enfermeiros era do sexo feminino, o comportamento contrário era verificado nos médicos. A aplicação do grad-CAM revelou que o modelo estava, de fato, baseando suas previsões em características faciais e estilo do penteado, uma vez que esse era um viés na base de dados. Após essa descoberta, os autores enriqueceram o conjunto de dados com uma representação mais equilibrada entre os gêneros para ambas as profissões, mitigando assim o viés identificado.

Em nossa pesquisa, aplicamos explicações visuais do grad-CAM para sondar potenciais vieses na base de dados UniMiB-SHAR. Ao analisar predições amostrais da arquitetura CNN2, identificamos eventos atípicos nas classes StandingUpFS, StandingUpFL, LyingDownFS e SittingDown, conforme ilustrado na Figura 6.8. Notou-se uma recorrência desse padrão tanto na abordagem SD quanto na SI para a arquitetura CNN2. Entretanto, dada a superioridade dos indicadores de desempenho da abordagem SD, optamos por utilizá-la como referência na avaliação dos resultados.

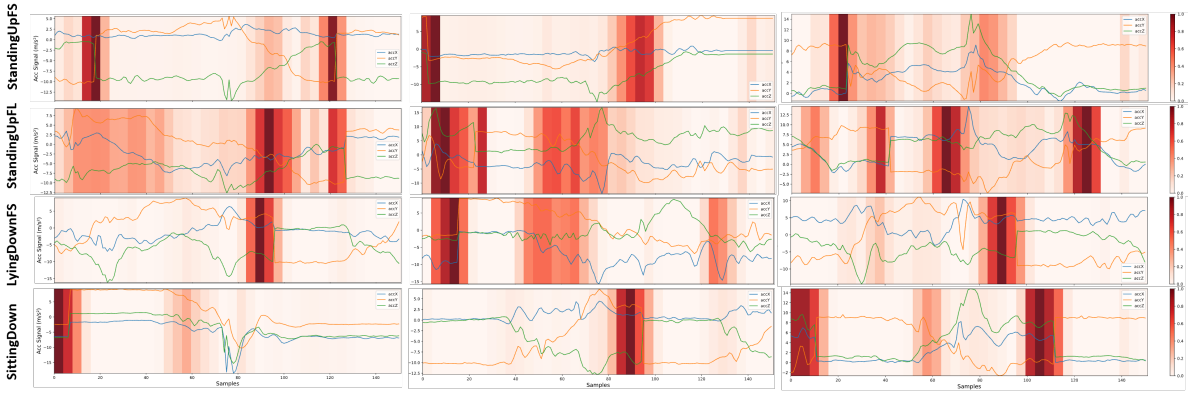


Figura 6.8. Potenciais problemas de viés no conjunto de dados foram identificados a partir de uma análise visual utilizando o grad-CAM nas amostras selecionadas. Nenhum limiar foi aplicado aos mapas de calor (*heatmaps*). Observa-se que a rede atribuiu excessiva importância às descontinuidades durante o processo de predição, o que possivelmente comprometerá a capacidade de generalização do modelo, uma vez que no mundo real o sinal não apresentará essa característica. Adaptado de (Aquino *et al.*, 2022).

Em estudos envolvendo aprendizado profundo, a precisão e consistência dos dados são essenciais. No entanto, a análise do conjunto de dados revelou várias inconsistências que podem impactar o desempenho dos modelos desenvolvidos. Tomando a classe *SittingDown* como estudo de caso, observou-se que a rede concentrou suas predições em segmentos ruidosos, isso pode ser notado pelo fato de que todas as regiões de importância nas amostras tomadas como exemplo, focaram nas descontinuidades do sinal.

Em uma análise superficial o leitor pode associar a descontinuidade a um movimento do sujeito durante a coleta de dados, porém, vamos analisar na Figura 6.8 a primeira amostra da classe *StandingUpFS*, onde há uma evidente descontinuidade nos eixos y e z. Esta amostra se encontra no canto superior esquerdo da imagem, observe a primeira região em destaque pelo grad-CAM, mais especificamente entre os segmentos 18 e 19, perceba que o sinal salta de -10 para $1,3$ no eixo y e decresce de $2,8$ para $-9,2$ no eixo z. Tal movimento abrupto é incompatível com a atividade de sentar-se, uma vez que, considerando uma frequência de amostragem de 50 Hz, tal movimento teria que ser executado em apenas $0,02$ s. Estas descontinuidades são recorrentes em algumas amostras das quatro classes mencionadas e elas impactam diretamente o aprendizado do modelo.

Várias causas podem estar por trás destas descontinuidades: problemas no sensor, falhas na aplicação de coleta de dados ou erros no processo de segmentação (*windowing*). Por exemplo, problemas de calibração do sensor podem gerar leituras inconsistentes. Irregularidades na aplicação de coleta podem resultar em uma frequência de

amostragem variável ou interrupções no registro devido a falhas no software, provavelmente este não é o caso que ocorreu, uma vez que os picos não são anomalias, o sinal de mantém no mesmo nível após a descontinuidade.

Quanto à segmentação, descontinuidades podem surgir durante a montagem do conjunto de dados, onde segmentos de uma amostra podem ter sido inadvertidamente combinados com segmentos de outra. Durante nossas observações, notou-se que estas descontinuidades podem ocorrer em todos os três eixos ou somente nos eixos y e z. Este pode ter sido o problema na geração da base de dados.

É importante notar que, ao invés de aprender características fundamentais das atividades, o modelo focou nestas irregularidades. Ou seja, a rede foi condicionada a reconhecer descontinuidades para diferenciar amostras destas classes, um viés que não existirá em cenários reais.

Ambas as arquiteturas foram influenciadas por estas descontinuidades, com a CNN2 mostrando-se mais susceptível. Além disso, tais problemas de descontinuidade também foram identificados nas classes de queda, impactando negativamente o aprendizado da rede.

Voltando a analisar ó desempenho obtido pela rede CNN2 com as abordagens SD e SI, com as Figuras 5.1 e 5.2 apresentadas previamente, notamos que a as classes mais problemáticas para SI, e as que possuíam um desempenho mais discrepante com relação a SD, possuíam vieses no conjunto de dados. Tomando como caso de estudo 3 das classes mostradas na seção anterior, as classes *SittingDown*, *LyingDownFS*, e *StandingUpFL*. Estas apresentaram as maiores diferenças de desempenho, ao observar a assertividade obtida nas matrizes de confusão.

Como no método SD as mesmas descontinuidades sobre os indivíduos aparecia durante o treino e a validação, o modelo conseguia utilizar essa informação para aumentar a sua performance. No entanto, no caso SI, as amostras dos indivíduos apresentavam descontinuidades diferentes nas vistas durante o treino, fazendo com que a rede obtivesse um desempenho inferior.

Com base nesta análise preliminar, é possível utilizar a discrepância de desempenho entre as abordagens SI e SD como um indicativo de possível viés na base de dados. A discrepância entre somente algumas das classes pode reforçar essa afirmação. Conforme visto, uma análise com grad-CAM pode ajudar a identificar esse viés e comprovar um possível problema com a base de dados.

6.4 Análises XAI com a aplicação do método t-SNE

Nesta seção, exploramos os resultados obtidos com a rede profunda CNN1 nos conjuntos de dados SHO e HAPT, empregando a visualização proposta por t-SNE através das características aprendidas. Os resultados são examinados e interpretados para enfatizar a eficácia do t-SNE na detecção de problemas de viés e identificação de amostras rotuladas incorretamente. A visualização ilustra ainda a capacidade de generalização da rede profunda em distinguir atividades e suas limitações quando aplicada a um conjunto de dados distinto. Por fim, a discussão oferece *insights* sobre como a visualização por t-SNE pode aumentar a compreensão das redes profundas em tarefas de HAR.

6.4.1 SHO

Após a fase de treinamento e a subsequente avaliação dos modelos por meio de métricas numéricas, propusemos a aplicação de técnicas de XAI com o objetivo de produzir representações visuais que esclareçam os mecanismos subjacentes ao processo de aprendizado dos modelos. Para tanto, procedemos com a extração dos vetores de características (*embeddings*, ou neste caso as características aprendidas) gerados pelo modelo imediatamente antes da aplicação da camada Softmax. Utilizando esses vetores como entrada, realizamos o treinamento de um modelo t-SNE para a redução de dimensionalidade. O resultado obtido foi um conjunto de vetores bidimensionais que encapsulam um resumo das características essenciais aprendidas durante o treinamento do modelo. Essa metodologia nos permitiu obter uma visualização intuitiva tanto do conhecimento adquirido pelo modelo a respeito das características intrínsecas dos dados quanto da maneira como os dados foram segmentados espacialmente, com base nas características assimiladas. A representação gráfica desses resultados é apresentada na Figura 6.9, ilustrando de maneira eficaz a distribuição dos dados em um espaço bidimensional de acordo com as aprendizagens do modelo.

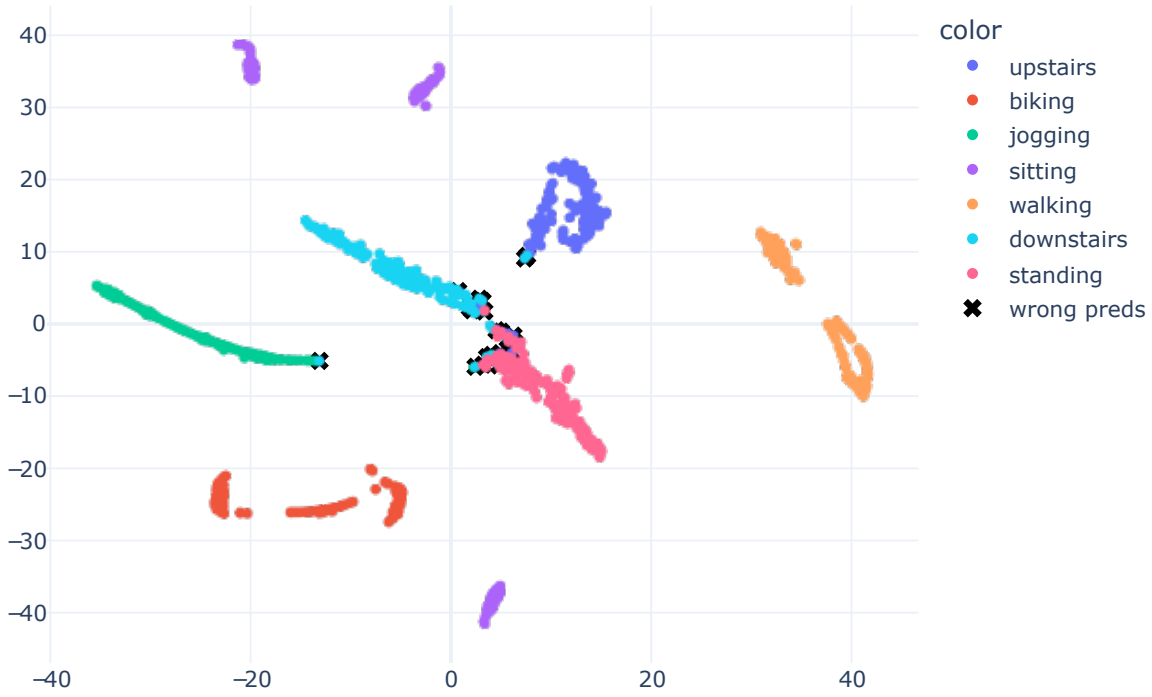


Figura 6.9. Gráfico de *Embedding* T-SNE obtido da arquitetura CNN1 com HAR usando a abordagem SI para o conjunto de dados SHO com o subconjunto de validação. Os erros de predição estão destacados com um x. Utilizamos perplexidade de 40 e realizamos 500 iterações. Adaptado de (Aquino *et al.*, 2023).

Os resultados visuais obtidos corresponderam aos resultados numéricos apresentados na Tabela 5.4, onde o modelo alcançou um desempenho excepcional com uma acurácia de $0,98$. No gráfico, os dados de diferentes classes foram dispersos. Confusão ocorreu entre atividades, como descer escadas e ficar de pé. Este resultado também foi evidente na matriz de confusão mostrada na Figura 5.4, onde a classe descer escadas alcançou menor precisão do que outras atividades.

Como retratado na Figura 6.9, marcas pretas em forma de x foram desenhadas em amostras com previsões incorretas para identificar os erros do modelo. A representação resultante permitiu verificar que as previsões incorretas foram prontamente observadas e explicadas. Por exemplo, examinamos a amostra mal classificada dentro do agrupamento de corrida, indicada por um x preto. A verdade fundamental para esta amostra era a classe descer escadas. Essa confusão surgiu devido às características aprendidas pelo modelo, que posicionou esta amostra perto do agrupamento de corrida. Isso pode ter ocorrido como resultado de uma limitação no modelo treinado, que pode ter aprendido características que não separaram os dados de forma ótima. Alternativamente, esta amostra poderia ter sido rotulada incorretamente. No entanto, a visualização obtida permitiu compreender por que o modelo cometeu um erro.

Na visualização, as amostras de descer escadas estavam mais dispersas, indicando que a classe descer escadas era a mais desafiadora no conjunto de dados. O modelo pode ser mais suscetível a prever esta classe em cenários do mundo real. Além disso, ao analisar os resultados do gráfico obtido, ficou evidente que havia subgrupos dentro da mesma classe, como sentar, que apresentava três subgrupos distintos dispersos pela visualização. Similarmente, até caminhar tinha dois subgrupos. Essas análises não eram possíveis apenas através de resultados numéricos, e a visualização apresentou várias oportunidades para explicar as previsões do modelo.

Uma observação intrigante foi feita ao analisar a atividade de sentar. Havia três agrupamentos, que correspondiam aos três diferentes sujeitos presentes no subconjunto de validação. Esse achado sugeriu que as características aprendidas podem ter o potencial de não apenas distinguir a atividade de sentar, mas também diferenciar os sujeitos.

Para destacar ainda mais as capacidades desta visualização, a Figura 6.10 exibe uma visualização usando PCA. O método aplicado foi similar ao t-SNE. Inicialmente, utilizamos a saída de *embeddings* do modelo, uma camada antes da classificação, como entrada para o PCA. Em seguida, aplicamos PCA para derivar apenas três componentes, que foram usados para gerar o gráfico. Calculamos apenas três componentes para demonstrar que não havia combinação de componentes para a qual a visualização resultante fosse tão informativa quanto a obtida com t-SNE.

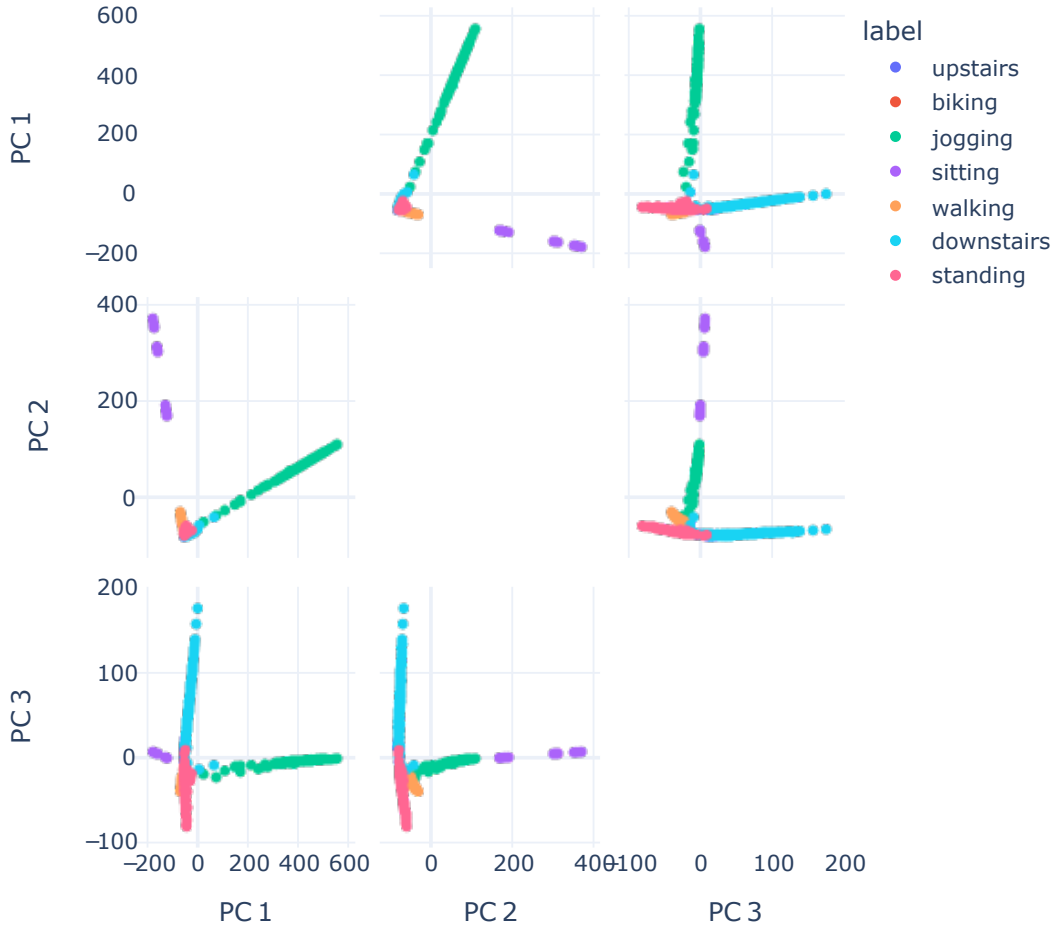


Figura 6.10. Visualização de *Embeddings* com PCA, combinando três componentes, 2 por 2. Adaptado de (Aquino *et al.*, 2023).

A representação na Figura 6.9 demonstra como o modelo aprendeu a separar os dados baseado nas características que adquiriu. Para analisar a distribuição dos dados brutos do acelerômetro, plotamos o gráfico exibido na Figura 6.11. Para gerar este gráfico, concatenamos os dados dos eixos X, Y e Z do acelerômetro e usamos como entrada para o algoritmo t-SNE. Nosso trabalho introduz a inovação de aplicar t-SNE uma camada antes da classificação do modelo, diferenciando-se da abordagem padrão de aplicar t-SNE em dados brutos, como visto em trabalhos anteriores.

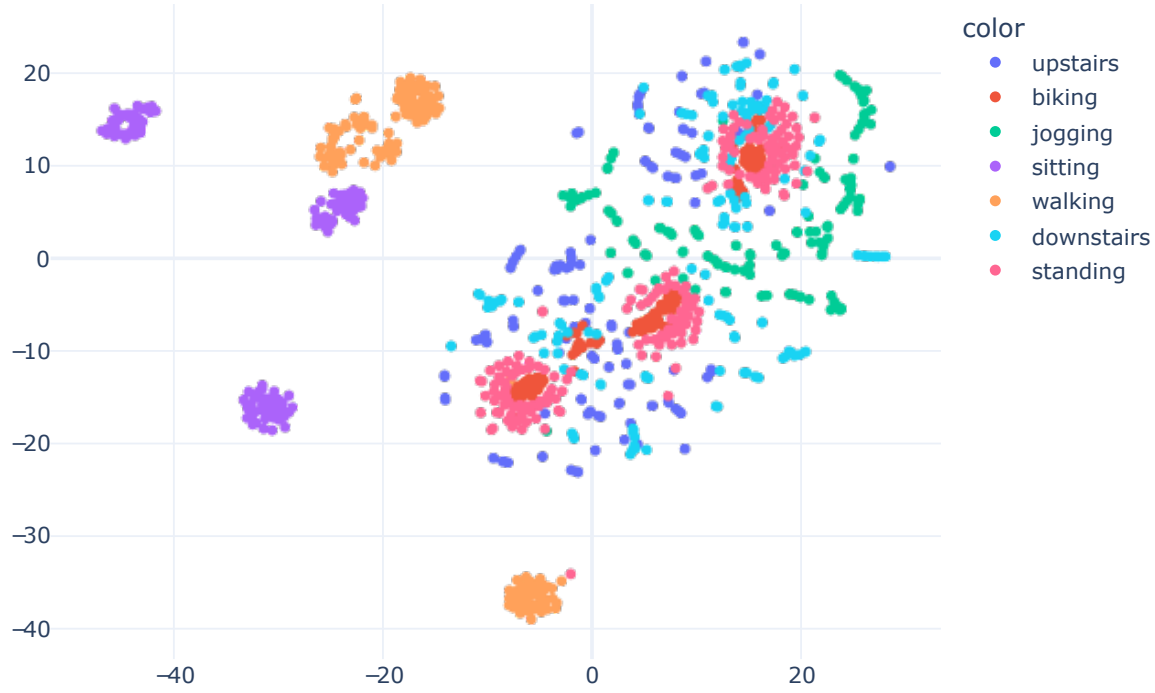


Figura 6.11. Visualização de dados brutos do acelerômetro SHO com t-SNE, concatenando informações X, Y e Z. Utilizamos perplexidade de 40 e realizamos 500 iterações. Adaptado de (Aquino *et al.*, 2023).

Ao visualizar o gráfico das características aprendidas durante o treinamento do modelo, tornou-se evidente que o modelo sofreu uma transformação em relação aos dados de entrada. Em sua forma original, os dados brutos apresentavam um desafio em diferenciar entre atividades humanas. No entanto, após o modelo aprender e destilar as características relevantes, a separação entre as classes tornou-se mais evidente e distinguível. Esta visualização ofereceu *insights* valiosos sobre o funcionamento interno do modelo e auxiliou na melhoria de seu desempenho.

6.4.2 HAPT

Para o conjunto de dados HAPT, com base nos resultados apresentados na Tabela 5.6, esperava-se uma confusão mais significativa entre as amostras em comparação com a representação obtida com o conjunto de dados SHO. Examinando a Figura 6.12, é possível observar uma menor separação entre amostras de diferentes classes. O principal problema estava nas classes de Transições Posturais (PT). Este resultado também foi antecipado ao observar a matriz de confusão na Figura 5.5.

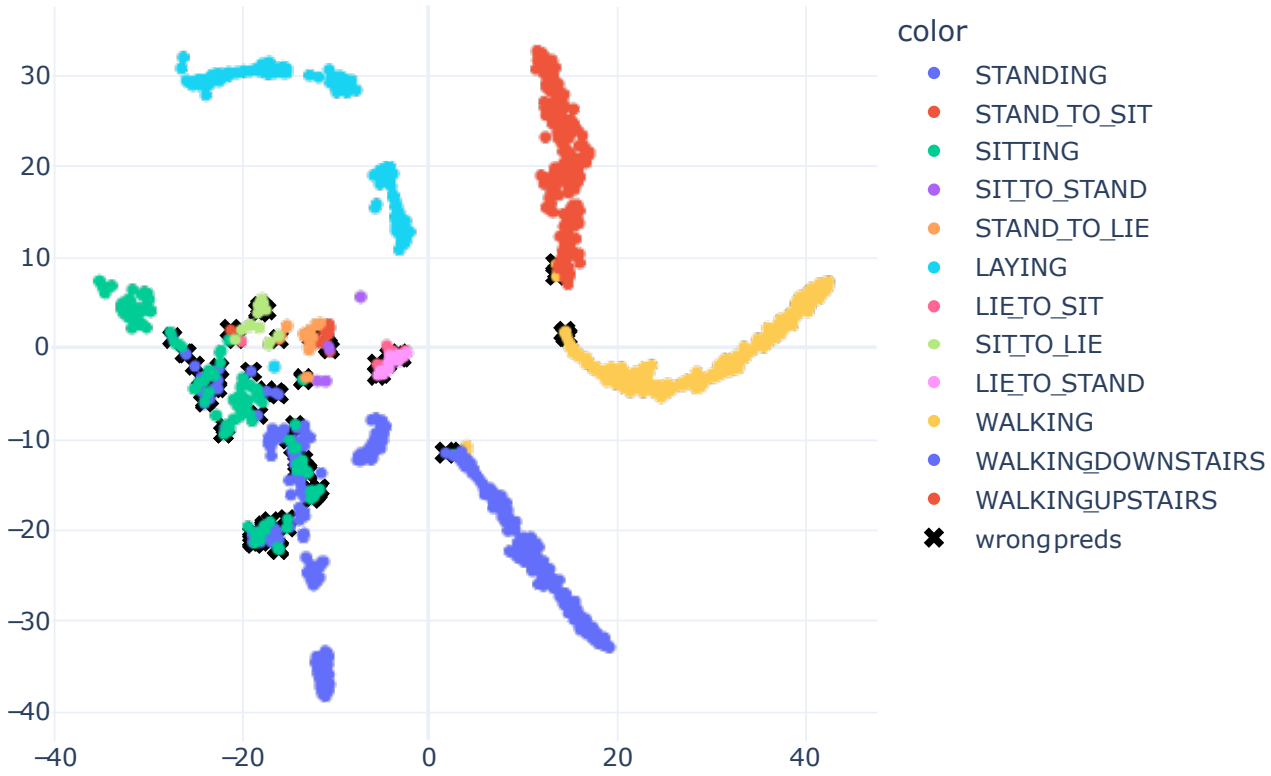


Figura 6.12. Visualização de dados brutos do acelerômetro para o conjunto de dados HAPT com t-SNE, no qual utilizamos perplexidade de 40 e realizamos 500 iterações. Adaptado de (Aquino *et al.*, 2023).

Para demonstrar que o modelo aprendeu características relevantes para as outras atividades, excluindo Transições Posturais (PTs), apresentamos a Figura 6.13. Outros trabalhos já consideraram essas PTs como um ou dois subgrupos. Alguns estudos usam PTs apenas para melhorar o desempenho no reconhecimento de outras atividades.

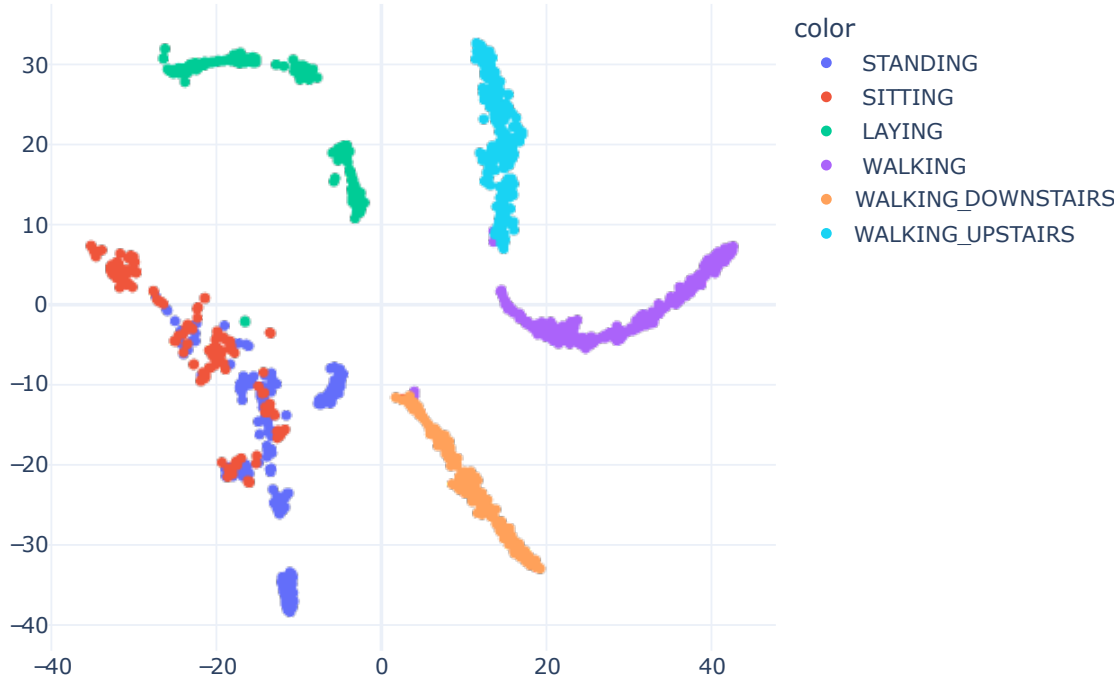


Figura 6.13. Visualização dos embeddings dos dados do conjunto HAPT com t-SNE. A visualização foi obtida com a abordagem SI, no qual foi excluídas as amostras das classes de PT. Utilizamos perplexidade de 40 e realizamos 500 iterações. Adaptado de (Aquino *et al.*, 2023).

O modelo encontrou um desafio familiar, semelhante ao conjunto de dados SHO, com a classe de estar em pé provando ser difícil de classificar. No entanto, em contraste com o conjunto de dados SHO, a confusão surgiu entre as classes de estar em pé e sentado. Análises adicionais revelaram a presença de subgrupos dentro das classes de deitar, estar em pé e sentar. Para melhor ilustrar a distribuição original dos dados, a Figura 6.14 exibe os dados brutos do acelerômetro usando t-SNE. O mesmo processo utilizado para gerar a Figura 4.4 foi aplicado ao conjunto de dados HAPT.

Figura 6.14. Visualização de dados brutos do acelerômetro para o conjunto de dados HAPT com t-SNE, concatenando as informações dos eixos X, Y e Z. Utilizamos perplexidade de 40 e realizamos 500 iterações. Adaptado de (Aquino *et al.*, 2023).

Como ocorreu com o SHO, utilizar dados brutos como entrada para o t-SNE foi um desafio, pois não havia características altamente relevantes, em contraste com a Figura 9, onde as características eram aquelas aprendidas pela rede neural e tinham relevância matemática. No geral, os modelos demonstraram sua capacidade de aprender características significativas, conforme evidenciado pelas visualizações t-SNE e métricas de desempenho obtidas em ambos os conjuntos de teste SHO e HAPT. A seguir, avaliamos se as características aprendidas em um conjunto de dados poderiam classificar com precisão as atividades humanas em outro conjunto de dados.

6.4.3 Características do SHO aplicadas ao HAPT

Para analisar a relevância das características, utilizamos uma rede treinada com o conjunto de dados SHO para extrair características do conjunto de dados HAPT. Para isso, realizamos uma predição com o modelo SHO, propagando as amostras do HAPT e considerando o resultado antes da camada de classificação. Este resultado foi utilizado como entrada para o algoritmo t-SNE.

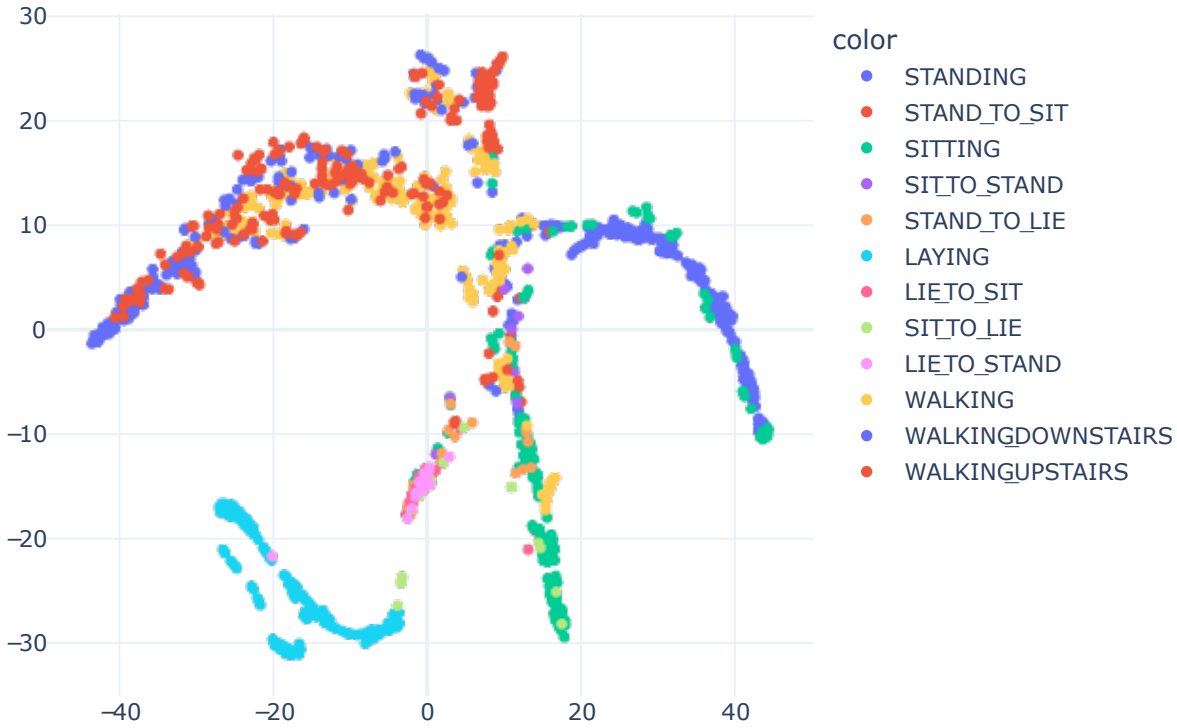


Figura 6.15. Gráfico de distribuição de características aprendidas com T-SNE obtido da arquitetura CNN1 treinada com a abordagem SI com o conjunto de dados SHO aplicado ao conjunto de dados HAPT, considerando todo o conjunto de dados. Utilizamos perplexidade de 40 e realizamos 500 iterações. Adaptado de (Aquino *et al.*, 2023).

A Figura 6.15 mostra a representação resultante. No entanto, as características aprendidas com o conjunto de dados SHO não se mostraram universais para serem úteis em outros conjuntos de dados. Neste caso, as características do SHO tornaram o padrão do conjunto de dados HAPT mais confuso do que os dados brutos. Embora a Figura 6.9 tenha distinguido com sucesso entre os tipos de atividades usando o conjunto de dados SHO como referência, esta abordagem produziu resultados insatisfatórios quando aplicada ao conjunto de dados HAPT.

6.4.4 Características do HAPT aplicadas ao SHO

Para analisar a relevância das características, utilizamos uma rede treinada com o conjunto de dados HAPT para extrair características do conjunto de dados SHO, conforme mostrado na Figura 6.16. Para isso, propagamos as amostras do HAPT através do modelo SHO e consideramos a saída antes da camada de classificação. Em seguida, usamos essa saída como entrada para o algoritmo t-SNE.

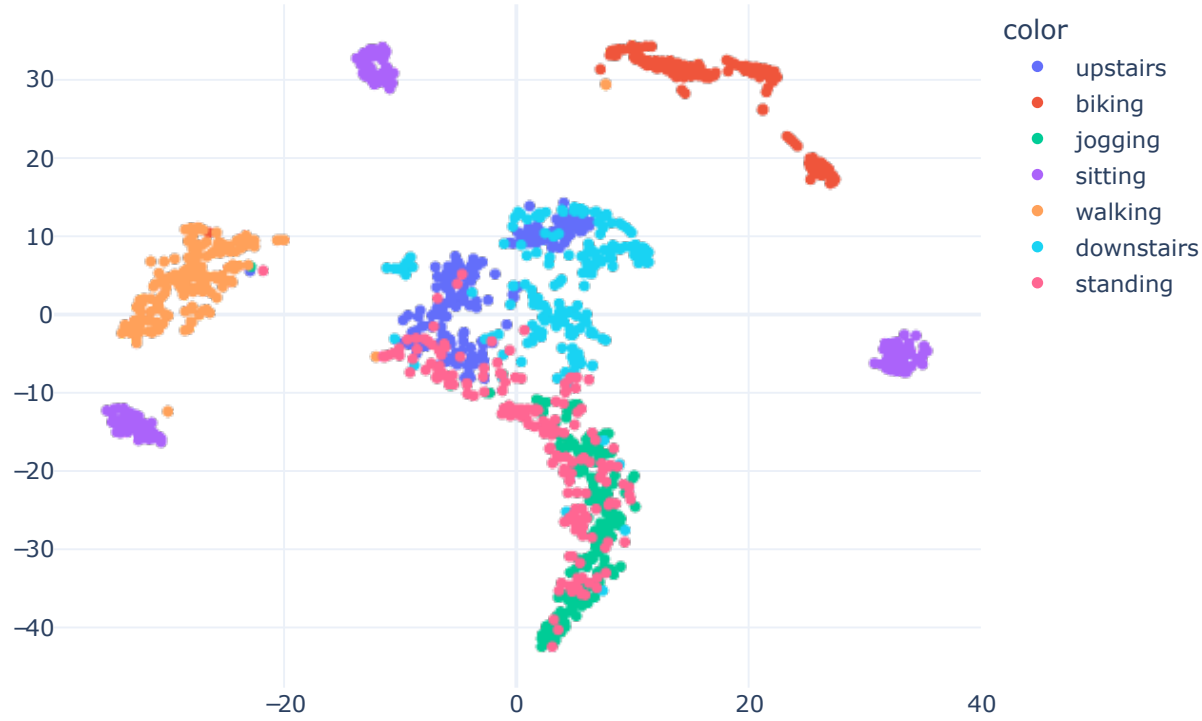


Figura 6.16. Gráfico de *Embedding* T-SNE obtido da arquitetura CNN1 treinada com a abordagem SI com o conjunto de dados HAPT aplicado ao conjunto de dados SHO. Utilizamos perplexidade de 40 e realizamos 500 iterações. Adaptado de (Aquino *et al.*, 2023).

Ao examinar o conjunto de dados HAPT, ficou evidente que a utilização das características aprendidas resultou em uma diferenciação mais precisa dos dados do que o uso dos dados brutos do acelerômetro. A Figura 6.14 mostra que o contraste entre as duas abordagens foi claro. As classes Sentar, Caminhar e Bicicleta foram bem separadas, ao contrário do que foi observado ao analisar a dispersão dos dados brutos, mostrada na Figura 6.11, onde a classe Bicicleta foi agrupada com outras classes. As classes Subir escadas, Descer escadas, Ficar em pé e Correr foram agrupadas em regiões próximas, mas ainda havia uma separação notável entre as amostras dessas classes.

7

Conclusão

Neste estudo, abordamos a importância e aplicabilidade das técnicas de XAI em modelos 1D CNNs no contexto de HAR e BUI. Observamos que, embora os modelos profundos, como aqueles baseados em redes convolutivas, tenham demonstrado um desempenho notável, a transparência de suas decisões muitas vezes é negligenciada. O foco deste trabalho foi propor uma abordagem que harmoniza a eficácia desses modelos com a transparência e compreensibilidade, através do emprego de métodos XAI e técnicas de ML avançadas.

Com a customização do grad-CAM para séries temporais em conjunto com 1D CNNs, foi possível não apenas elucidar as decisões tomadas por esses modelos, mas também fornecer uma ferramenta poderosa para a identificação e correção de vieses. As visualizações obtidas por meio das técnicas de XAI proporcionaram *insights* profundos sobre o processo de aprendizado, revelando a forma como os modelos interpretam e classificam os sinais de entrada. Especialmente, a análise das correlações entre HAR e BUI por meio destas visualizações expôs como variações nos dados podem influenciar o viés de aprendizado, uma contribuição direta aos objetivos de investigar e propor métodos para a interpretação e explicação de decisões em 1D CNNs aplicadas a HAR.

A técnica t-SNE, adaptada para visualizar o aprendizado dos modelos, permitiu uma compreensão mais ampla das características aprendidas, transcendendo a análise de amostras individuais para oferecer uma visão macro das capacidades de aprendizado. Essa abordagem alinha-se com o objetivo de explorar métodos que não apenas avaliam o desempenho numérico dos modelos mas também a qualidade e a interpretabilidade das características que aprendem, orientando a seleção de modelos que são robustos tanto quantitativa quanto qualitativamente.

No entanto, a abordagem apresentada, especialmente a utilização do t-SNE, enfrenta limitações que ressaltam a necessidade de pesquisas futuras. Isso inclui a explo-

ração de sua aplicabilidade a diferentes tipos de dados e algoritmos de DL, um reflexo do objetivo de investigar técnicas de visualização adaptáveis a uma ampla gama de contextos de HAR. As limitações observadas não diminuem a contribuição deste trabalho, mas sim destacam a complexidade do desafio de tornar os modelos de aprendizado profundo transparentes e explicáveis.

Através deste trabalho, contribuímos para o campo de HAR e BUI ao propor uma metodologia que melhora a análise de desempenho dos modelos profundos não só por seus resultados numéricos mas também pelo entendimento de seu aprendizado e processos decisórios. Além disso, as arquiteturas CNN1 e CNN2 foram equiparáveis ao estado da arte de aplicações de HAR, podendo estar muito próximo ou até mesmo ser o estado da arte para os conjunto de dado analisados.

Este trabalho destaca a importância de avançar além da busca por alto desempenho numérico, enfatizando a necessidade de compreender como modelos profundos tomam decisões. Ao fazê-lo, abrimos novas avenidas de pesquisa que prometem tornar as aplicações de HAR mais confiáveis, explicáveis e, por fim, mais úteis em cenários do mundo real. Em suma, a pesquisa realizada enfatiza a relevância da transparência e da compreensibilidade na era do aprendizado profundo, pavimentando o caminho para um futuro onde modelos de IA são não apenas poderosos mas também plenamente interpretáveis e justos.

Em síntese, a contribuição deste trabalho não se limita à proposta de técnicas inovadoras, mas também à demonstração prática de como o XAI pode enriquecer, aprimorar e tornar mais transparente a aplicação de modelos DL em HAR, BUI e em outras aplicações baseadas em séries temporais.

Para futuros pesquisadores como lições aprendidas destacam-se alguns pontos:

- A diferença de performance entre as abordagens SI e SD pode indicar um viés no dataset.
- A visualização grad-CAM pode ser aplicada para observar a tomada de decisão de uma única amostra, permitindo a identificação de possíveis pontos fracos dos modelos.
- É possível alcançar uma performance melhor ou equivalente utilizando os sinais brutos do acelerômetro na entrada de modelos profundos, em vez de considerar apenas as características extraídas.
- Em um contexto onde é necessário avaliar o conjunto de dados como um todo, a visualização proposta com o t-SNE pode ajudar a entender melhor o comportamento e a tomada de decisão dos modelos treinados.

- Com base nos testes realizados, as CNNs são um bom ponto de partida como estratégia de rede a ser utilizada nas bases de dados. As arquiteturas CNN1 e CNN2 propostas podem ser utilizadas como ponto de partida para um bom desempenho.

- A visualização com t-SNE proposta pode ser utilizada para avaliar a qualidade das características aprendidas pelos modelos, mesmo em bases de dados diferentes das que foram inicialmente treinadas.

- A visualização com grad-CAM pode ser utilizada para identificar viés em modelos, enquanto a visualização com t-SNE pode ser utilizada tanto para a identificação de viés em datasets quanto em modelos.

De forma geral, este trabalho propõe uma forma mais eficaz de analisar o desempenho dos modelos profundos aplicados a HAR. De forma a escolher um melhor modelo não apenas pelo seu desempenho numérico, mas também pela análise do seu aprendizado e tomada de decisão.

Este estudo abre caminho para diversas vias de pesquisa, incluindo as seguintes:

- Embora a última camada seja a que mais contribui para a tomada de decisão do modelo, é relevante analisar por meio do grad-CAM a contribuição de todas as outras camadas convolutivas? Como poderíamos ponderar as informações mais importantes?

- Qual estratégia de validação resulta em uma melhor separação das classes em um conjunto de dados de Reconhecimento de Atividades Humanas (HAR), Inter-Sujeito (SI) ou Divisão de Sujeito (SD)?

- Se dados adicionais sobre a posição dos sensores fossem incorporados a base de dados SHO, isso melhoraria a separação das classes HAPT?

- É possível reverter o processo? Partindo das coordenadas de incorporação do t-SNE, é possível recuperar os dados brutos? Se sim, isso poderia ser utilizado para aumento de dados no conjunto de dados?

- Como as incorporações derivadas de outros modelos de redes profundas, como LSTM, BiLSTM, GRU, MLP e modelos híbridos, se comportam?

- Como a técnica de visualização proposta deve ser empregada para evitar confusões de classe, modificando a arquitetura para melhorar e facilitar a diferenciação de classes desafiadoras?

O presente trabalho resultou na publicação de 2 artigos científicos diretamente relacionados com o tema dessa pesquisa e 4 outros relacionados ao tópico de reconhecimento humano de atividades, conforme abaixo:

- Explaining One-Dimensional Convolutional Models in Human Activity Recognition and Biometric Identification Tasks (Aquino *et al.*, 2022).
- Explaining and Visualizing Embeddings of One-Dimensional Convolutional Models in Human Activity Recognition Tasks (Aquino *et al.*, 2023).
- A New Approach for Detecting Activities of Daily Living Detection Using Smartphone and Wearable Sensors (Costa *et al.*, 2021).
- Gait analysis: determining heel-strike and toe-off events (Costa Filho *et al.*, 2021).
- Human Activity Recognition from Accelerometer Data with Convolutional Neural Networks (de Aquino e Aquino *et al.*, 2020).
- Human activity recognition from accelerometer with convolutional and recurrent neural networks (Serrão *et al.*, 2021).

Apêndice

Artigo 1 - Explaining One-Dimensional Convolutional Models in Human Activity Recognition and Biometric Identification Tasks

sensors

MDPI

Article

Explaining One-Dimensional Convolutional Models in Human Activity Recognition and Biometric Identification Tasks

Gustavo Aquino, Marly G. F. Costa and Cicero F. F. Costa Filho *

R&D Center in Electronic and Information Technology, Federal University of Amazonas, Manaus 69077-000, Brazil; gustavoaqui@gmail.com (G.A.); mcosta@ufam.edu.br (M.G.F.C.)
* Correspondence: ccosta@ufam.edu.br

Abstract: Due to wearables' popularity, human activity recognition (HAR) plays a significant role in people's routines. Many deep learning (DL) approaches have studied HAR to classify human activities. Previous studies employ two HAR validation approaches: subject-dependent (SD) and subject-independent (SI). Using accelerometer data, this paper shows how to generate visual explanations about the trained models' decision making on both HAR and biometric user identification (BUI) tasks and the correlation between them. We adapted gradient-weighted class activation mapping (grad-CAM) to one-dimensional convolutional neural networks (CNN) architectures to produce visual explanations of HAR and BUI models. Our proposed networks achieved 0.978 and 0.755 accuracy, employing both SD and SI. The proposed BUI network achieved 0.937 average accuracy. We demonstrate that HAR's high performance with SD comes not only from physical activity learning but also from learning an individual's signature, as in BUI models. Our experiments show that CNN focuses on larger signal sections in BUI, while HAR focuses on smaller signal segments. We also use the grad-CAM technique to identify database bias problems, such as signal discontinuities. Combining explainable techniques with deep learning can help models design, avoid results overestimation, find bias problems, and improve generalization capability.

Keywords: human activity recognition; grad-cam; convolutional neural networks; biometric user identification; deep learning; accelerometer data; explainable AI

Citation: Aquino, G.; Costa, M.G.F.; Costa Filho, C.F.F. Explaining One-Dimensional Convolutional Models in Human Activity Recognition and Biometric Identification Tasks. *Sensors* **2022**, *22*, 5644. https://doi.org/10.3390/s22155644

Academic Editor: János Abonyi

Received: 7 July 2022
Accepted: 25 July 2022
Published: 28 July 2022

1. Introduction

Smartphones and wearable devices provide unprecedented opportunities to monitor human physiological signals, creating possible applications in different areas such as activity tracking, ambient assisted living, healthcare, and others [1]. The global wearable-device market is projected to grow at 11.3% by 2025, resulting in a revenue of $62.82 million [2]. In recent decades, several papers have been published, with different techniques and solutions in HAR, using the wide variety of sensors incorporated into a smartphone, such as accelerometers, gyroscopes, microphones, GPS units, and others. Mobile devices use these sensors to register data and apply personal assistants to monitor and help the users in their exercise and activity routines, making them healthier. For example, it is known that simple physical activities such as sitting, walking, running, climbing, or descending stairs can help reduce the risk of chronic diseases such as obesity, diabetes, and cardiovascular diseases [3]. With HAR techniques, it is possible to create an assistant that can monitor a user's daily activities and improve his or her health conditions by making recommendations based on the detected activities and encouraging the user to exercise more.

Another use of HAR is in ambient assisted living. As the population's average age worldwide has increased considerably, healthcare needs of the elderly, such as physical rehabilitation, physical support, and home care [4], have been increasing. In this context, many studies have been done on fall detection [5–7]. These systems can monitor older people and help decrease their medical expenses, increase their independence, and improve their quality of life.

HAR can be implemented using a variety of sensors. We can use inertial measurement units (IMU), cameras, microphones, and GPS, among others. Although studies use cameras in HAR, this is not a predominant approach due to user privacy concerns. Among wearable devices, smartphones are the most widely used [4]. Smartphones are affordable and widespread. Furthermore, they are already integrated into the daily routine, making them even more interesting for HAR. With them, no additional device is needed. The smartphone is one of the first devices people come in contact with when they wake up [8]. Developing a smartphone app is more accessible than other wearables, such as smartwatches. There are many different open-source repositories to use as a base, either for collecting data from onboard sensors, running and inferring machine learning models locally, or sending the data to a cloud architecture. In addition, the smartphone typically has more processing power than other devices, which helps it to collect data efficiently and execute deep learning models.

The IMU is the most widely used in HAR among the available sensors. The IMU is an electronic device that can report position variations and provide time series data. It is typically an integrated sensor package that includes an accelerometer, gyroscope, and magnetometer. The tri-axial accelerometer can measure the rate of change of the body's velocity through the x-, y-, and z-axis at its local position. It is the most commonly used sensor in HAR [1,4]. We can use the accelerometer data alone to perform HAR, or we can use it in combination with a gyroscope or a magnetometer to improve performance in HAR. It is not typical to develop HAR with only a gyroscope or magnetometer. An essential factor for activity recognition is the sampling rate, defined as the number of readings taken of a piece of data per second, usually expressed in hertz.

Deep learning (DL) is a field of machine learning (ML). Recently, DL has outperformed ML in many applications, such as time series classification, image recognition, speech recognition, object detection, and natural language processing. In object detection, for instance, DL can improve energy efficiency in intelligent buildings. Recognizing the number of people in a room maximizes energy expenditure and guarantees thermal comfort [9]. Traditional ML requires expert human knowledge to extract the relevant attributes from sensor signals and much human effort. For example, in HAR, there is a standard methodology known as Activity Recognition Protocol (ARP), which consists of five steps: preprocessing, segmentation, feature extraction, and classification and evaluation [4,10]. In classic ML, the feature extraction step is done manually by an expert who has studied the essence of the problem and defined the most relevant pieces of information to categorize the problem.

On the other hand, with DL, we can use a neural network (NN) to extract the best features and classify them during its training process, simply giving raw data as input. Convolutional neural networks (CNNs) are a kind of deep architecture. CNNs are so named because of the convolutional layer in their structure, responsible for automatically extracting features during training. In HAR, CNNs can receive both raw data and handcrafted features. Mekruksavanich et al. [11] used CNNs as their base model for biometric user identification (BUI). The model had tri-axial accelerometer and tri-axial gyroscope data as inputs. Their system was composed of two cascade classifiers. The first one was used for activity classification, and the second one performed recognition of the individual.

A significant problem in using DL is the difficulty of interpreting the model. There is a trade-off between performance and simplicity or interpretability. Classic ML models, for example, decision trees, are highly interpretable and explainable but are not performing in complex scenarios. Selvaraju et al. [12] introduced grad-CAM (gradient-weighted class activation mapping), a technique for producing visual explanations for CNN-based models. With grad-CAM, we can better understand model limitations, discerning a "stronger" deep network from a "weaker" network, even when their performance metrics point to similar results. So far, in DL systems, we have sacrificed explanation capability for better performance. However, with grad-CAM, this paradigm has started to change.

HAR plays an essential role in people's daily lives due to wearables' popularization. The evolution in these devices' ability to extract high-level information about the user's

routine is due to the scientific community's efforts on the topic of HAR. Advances in DL have also played a vital role in this area. DL helps create high-performing models. However, it is not easy even for AI developers to understand how the model makes decisions. When you work with image data, some techniques can assist the designer in understanding the models' decisions, such as grad-CAM or object recognition models, where the region responsible for the decision making is clearly expressed by bounding boxes. However, this is not extensively explored in HAR and BUI, with time series data from inertial sensors. Most often, the articles only look at performance metrics to estimate the potential and capability of the models. The lack of references using XAI methods in deep learning based on one-dimensional convolutional models applied to HAR and BUI tasks creates difficulties for the AI developer in building a robust model in real-world scenarios.

This paper shows how to generate visual explanations that help to understand the learning of the trained CNN models on both HAR and BUI tasks. We propose a new CNN architecture capable of performing HAR and BUI. We have used the public dataset UniMiB-SHAR [7], which has 17 classes, including daily activities and falls, performed by 30 individuals. To perform the HAR, we trained and compared classifiers with two validation strategies. In the first, known as subject-dependent (SD), we used a hold-out split of 70/30 for training and validation. In the second, known as subject-independent (SI), we used 21 subjects for training and 9 subjects for validation, reaching 0.978 and 0.755 accuracy for each strategy, respectively. We also developed BUI models for each of the 17 activities, differentiating each user who performed the activity, achieving 0.937 average accuracy. Our experiments also revealed a linear correlation with a Pearson coefficient of 0.77 between the results obtained in BUI and the classifier's performance that executes activity recognition with the SD strategy. In SD, a network learns not only the physical activity but also the signature of the individuals who perform the activities. In addition, we used the grad-CAM technique to produce visual explanations that help understand and explain the convolutive models developed, by examining each architecture to identify possible generalization problems. Finally, we show how the models were affected by the biases of the database.

Specifically, our contributions are summarized as follows:

(1) We propose a new CNN architecture capable of being used to perform HAR and BUI;
(2) We show a relationship between the network's performances of HAR with SD and BUI;
(3) We propose a methodology to analyze the learning of one-dimensional CNNs in both HAR and BUI models, using the grad-CAM technique to produce visual explanations that highlight what the model took into account to make a prediction;
(4) We use the visual explanations to identify bias problems in the database.

To the best of our knowledge, this is the first study to present a way to generate visual explanations for one-dimensional convolutional networks applied to HAR and BUI tasks.

The rest of the paper is organized as follows: Section 2 introduces the standard protocol used in developing activity recognition applications. Section 3 presents the related works that perform both HAR and BUI. Section 4 presents the methodology, introducing the dataset, CNN architectures, and the grad-CAM technique. Section 5 presents the results achieved for each experiment performed. Finally, in Section 6, we discuss the results.

2. Activity Recognition Protocol

Most HAR applications apply the standard Activity Recognition Protocol (ARP) [1,4,10,13–15]. Basically, ARP consists of five steps: data acquisition, preprocessing, segmentation, feature extraction, and classification and evaluation, as follows.

2.1. Acquisition

Responsible for acquiring data from the sensors. At this stage, an application is used to acquire and store the data from the activities, following previously established acquisition protocols. Many works report using a camera or microphone to help label the data [7,14]. The sensor most often used for data capture is the accelerometer since it can measure

the direction of movement over time. Many sampling rates have been considered in the literature, but 50 Hz is the most widely used [4].

2.2. Preprocessing

Preprocessing can be performed at the time of acquisition or after it. Since the preprocessing method can affect the model performance, executing this step after the acquisition is more common. In preprocessing, data science techniques, digital signal processing (DSP), and machine learning (ML) techniques can improve data quality, identify outliers, and remove noise. With DSP, we can correct sampling problems that occurred during signal acquisition. ML techniques make it possible to analyze the data distribution. With data science, we can treat missing values.

At this stage, some authors apply filters to correct problems caused by the acquisition process, such as a low-pass filter with a cutoff frequency ranging between 0.1 and 0.5 Hz, in order to isolate and remove the gravitational component because it is a bias that can influence the performance of the model [16].

2.3. Segmentation

In this step, the data is processed in smaller segments called windows. To facilitate learning in the model, data quality is further improved. Different activities performed by individuals may have different durations. For example, in the case of the physical activity walking, each individual can take a different period to walk a specific distance. Compared to the walking activity, the jumping activity usually occurs in a shorter time period. With segmentation techniques, we can balance the time windows of the samples, considering each subject's characteristics and activity.

In order to make the raw data suitable for use by the model, we start by choosing an optimal window size to recognize all activities. All samples in the dataset should have the same time window. The typical window size is 3 s, used in 56% of recent studies [4]. Next, we have to define the method for windowing the data. The most commonly used techniques are event-defined windowing and sliding windowing. With the former, the windowing process is done around a target event to be detected, such as a spike. The second approach divides the data into fixed-size windows at a constant step. This process may or may not overlap between the samples, meaning that a part of the present sample can include a part of the previous sample. It is common to use a 50% overlap between the samples [4].

2.4. Feature Extraction

Feature extraction is defined as a process of obtaining information from the signal using mathematical relationships present in it. There are two approaches to this step: handcrafted features and learned features. With the handcrafted approach, we rely on the knowledge of an expert in the area to obtain relevant mathematical relationships capable of differentiating the classes of the problem. In contrast, with the learned features approach, we use machine learning techniques to obtain these features based on data correlations. Although the use of learned features, with raw data input to the networks, has been widely accepted by the scientific community, when working with images, in time series problems such as in HAR, many works are still using handcrafted features [1,4,13]. This is because a deep network requires a higher computational cost than a shallow network. Since ample computational power is unavailable in smartphones and wearables, classical approaches are preferred in most cases, and they do not achieve high performance with raw data.

Handcrafted features can be divided into time, frequency, and symbolic categories. In the time domain, these features are obtained by statistical calculations. Some examples of statistical features are the mean and the standard deviation. Frequency domain features are calculated after applying the Fourier transform to the data. Some examples of characteristics in the frequency domain are the entropy and the sum of the spectral power components.

Symbolic features are obtained after a discretization process. Some examples of symbolic features are skewness and kurtosis.

For the learned features approach, some studies are using DL and principal component analysis (PCA) [15,17–20]. DL algorithms learn features during their training stage. For example, CNNs perform the mathematical convolution operation during their learning process. These nets are trained based on the backpropagation algorithm, which can modify both the kernel coefficients, in the feature extraction layers, and the weights of the classification layer. The kernel coefficients are used to obtain the feature maps. In CNNs, the feature maps are the learned features. Using PCA, which aims to reduce data dimensionality, a set of orthogonal features are extracted from the data, called principal components. These components are the learned features of this approach and can be used as inputs to a classifier. Another way to obtain learned features is with autoencoder networks. The autoencoder is composed of an encoder and a decoder. This neural network architecture seeks to compress the data and reconstruct it from the compressed representation. It is possible to obtain the primary data information, capable of representing it completely, known as latent space. The encoder reduces the dimensionality of the data, while the decoder reconstructs the signal. We can develop autoencoders with CNNs or dense layers. In autoencoders, the latent space is the learned features.

2.5. Classification and Evaluation

We have two possible approaches when working on classification systems, model-driven or data-driven [4]. With the model-driven paradigm, we have a solid idea of how the physical system works, and we try to replicate it manually using composition rules that can represent the problem as an equation. With the data-driven paradigm, we can use machine learning to find links and correlations based on many variables and data. The latter is most often adopted in HAR.

In ML and DL, we use mathematical and statistical techniques to build intelligence capable of solving a problem. Through these techniques, a model does not have to be explicitly programmed. They should be able to learn from the data. Classification is one of the possible applications in this area. Many classifier algorithms use classical ML, such as naive Bayes, support vector machines, decision trees, random forests, and many others. Some classifiers that use DL are the multilayer perceptron, CNNs, residual neural networks, and long-term recurrent networks. The choice of the classification algorithm can dramatically influence classification performance. Generally, classical ML classifiers need the sensor data to be converted into a high-level representation to solve the problem. In other words, these models cannot receive the raw data. The raw input must be converted into representative information to teach a network; this is why feature extraction is essential for classical ML classifiers. They cannot receive raw data because it contains too much unnecessary information and noise. Some classical ML algorithms cannot solve nonlinear problems. When we are developing our classifier, we do not have enough knowledge about whether this will be an "easy" or "hard" task for a neural network. DL systems can solve easy problems and also handle complex problems better. However, they require a large amount of data, while classical ML can handle few data and less complex tasks better. In DL, the complexity of a network is related to the depth of the architecture, i.e., the number of neurons and layers they have.

Once the classification algorithm is chosen, we must choose a validation strategy to divide our dataset into training and testing subsets. This step is needed because ML and DL models must be evaluated with both seen and unseen data, to see if the model has learned to generalize to different instances of the same problem. There are three validation approaches: subject-dependent (SD), subject-independent (SI), and hybrid. The SI strategy does not use end-user data during training. In contrast, the SD and hybrid approaches use this data during the training process. However, the hybrid approach tries to use end-user data in smaller proportions.

After deciding on the training algorithm and its validation strategy, we must choose a set of metrics that evaluate the developed model's performance during training and validation. The metric value achieved in the training subset does not represent the model's performance. We must evaluate the model's capability based on data not seen in the training, so the metric values achieved in validation are more significant. The most commonly used evaluation metrics in HAR are accuracy, recall, precision, and F1-score, as follows in Table 1.

Table 1. Most commonly used metrics in HAR systems.

Metric	Equation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Recall	$\frac{TP}{TP + FN}$
Precision	$\frac{TP}{TP + FP}$
F1-score	$2 \times \frac{Precision \times Recall}{Precision + Recall}$

All metrics are based on true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The TP, TN, FP, and FN concepts are grounded in binary classification. However, they can be extended to a multiclass classification with a one-vs-all strategy. This strategy considers the target class positive, and all other classes are grouped into a negative class. Accuracy is the most widely used metric and represents the overall assertiveness percentage. Accuracy is a simple metric to understand, but when we have a dataset with an imbalance of the classes, it does not indicate the real performance of the classifier. The precision metric shows how the model handles precision in predicting positive samples. Our model recognizes TP samples very well if we have high precision. The recall metric shows how well our model rejects false negatives. This metric is used when there is a high-cost association with an FN. The F1-score measures a balance between precision and recall. This metric is essential when we have an imbalanced class distribution.

3. Related Works

This section will discuss related studies in the HAR area, focusing on those that used the UniMiB-SHAR dataset. Next, we discuss the studies that perform BUI based on previous activity recognition, and finally, the grad-CAM method will be presented, together with the studies that are using the technique.

3.1. Related Studies in HAR

The first study to implement HAR dates back to the 1990s. Since then, many techniques and frameworks have been proposed. Many sensors can be used in HAR, such as cameras, piezoelectric, GPS, microphone, and IMU.

CNN architectures have been widely used in HAR. They are considered robust due to their scale and transition invariance. Most CNN architectures are encoder-like, i.e., the data loses dimension as it moves into the deeper layers. This contributes to obtaining high-level semantic meanings. These architectures can extract representative features as good as, if not better than, handcrafted methods [15]. They can receive multidimensional data as inputs. For example, a CNN can receive as input a 9-dimensional signal consisting of a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer. They are not limited to working only with raw data. These nets can also receive handcrafted features or learned features from other deep networks. Dong et al. [21] propose HAR-Net, a CNN architecture that can receive handcrafted features on its input. Lee et al. [22] calculate the vector sum of the accelerometer in the x-, y-, and z-axis to obtain a magnitude signal they use as input to a CNN, achieving better performance than a random forest algorithm. Ronao et al. [23] propose a CNN capable of receiving 6-dimensional data as its input, composed of a tri-axial accelerometer and a tri-axial gyroscope. The network achieved 94% accuracy.

Many recent papers have used the UniMiB-SHAR database, as shown in Table 2. All articles used 17 classes and have only raw accelerometer data in the classifier input.

Table 2. State-of-art studies using the UniMiB-SHAR dataset; N × M—N is the input size and M is the input dimension; SD—subject-dependent; SI—subject-independent; CV—cross-validation; LOSO—Leave-one-subject-out.

Author	Model	Input (N × M)	Validation Strategy	Best Performance
Micucci et al. [7]	KNN and SVM	51 × 3 and 151 × 3 samples	SD: CV-5 and SI: LOSO	SD with 151 × 3—0.830 of MAA * with KNN SI with 151 × 3—0.547 of MAA with KNN
Mukherjee et al. [24]	Ensemble of CNN models	151 × 3 samples	SD: CV-5	SD: 0.926 of accuracy
Tang et al. [25]	CNN	151 × 3 samples	SD: Hold-out 70/30	SD: 0.775 of weighted F1-score
Serrão et al. [6]	GRU and CNN-2D	151 × 3 samples	SD: CV-5 and SI: LOSO	SD: 0.955 of MAA and 0.978 of accuracy with GRU SI: 0.714 of MAA and 0.798 of accuracy with CNN-2D
LV et al. [26]	ConvLSTM	151 × 3 samples	SD: CV-5 and SI: LOSO	SD: 0.953 of MAA and 0.973 of weighted F1-score SI: 0.784 of weighted F1-score
Teng et al. [21]	Resnet (CNN)	151 × 3	SD: Hold-out 70/30	SD: 0.806 of weighted F1-score and 0.809 of accuracy
Cheng et al. [27]	CNN	151 × 3 samples and 151 × 1 (magnitude)	SD: Hold-out 70/30	SD: 151 × 3: 0.886 of accuracy and SD: 151 × 1: 0.7731 of accuracy

* MAA = Mean Average Accuracy.

Some papers focus on a new neural network architecture or processing approach, claiming higher performance than their baseline [6,7,24,26]. Usually, these studies show either a benchmark with previous papers that have used UniMiB-SHAR or show the performance of its technique on different datasets. Some articles propose an improvement in the optimization method used to train the deep network or in its internal structure, comparing the performance of their method with traditional approaches [21,25,27].

Mukherjee et al. [24] propose three classification models, named CNN-Net, Encoded-Net, and CNN-LSTM. Each of these models is built using one-dimensional CNNs as a basis. Their input is a two-dimensional matrix obtained from a windowing application with signal overlap. Each neural network performs a vote to decide a class. The class with the highest vote is the predicted class. Micucci et al. [7] used the raw data to input KNN and SVM algorithms. This particular study was the first baseline, as this was the author of the UniMiB-SHAR database. Their study is intended to introduce the dataset and show with a simple implementation that it is possible to perform activity recognition with machine learning algorithms. Tang et al. [25] propose a low computational cost CNN using Lego filters. This work is concerned with the limitation we have when applying deep learning techniques to devices with low computational cost, such as wearable devices. A set of lower-dimensional filters are used in a similar way to Lego bricks. The strategy used showed that it is possible to reduce memory and computation costs using CNN. The authors claim that the model is small, fast, and accurate. LV et al. [26] present a hybrid deep learning network with ConvLSTM architecture. This study joins two deep network architectures, CNN and LSTM networks, to make better use of information extracted by CNN in the temporal domain, also inserting a dense link module to improve the information flow between network layers and a multilayer feature aggregation module to extract features along with spatial domains. The authors also aggregate the features obtained from each convolutional layer according to their importance at different positions in the signal. Teng et al. [21] propose both a new training method and a new architecture for the residual neural network. Residual networks are a specific type of CNN, which generally do not follow a sequential architecture, aggregating information from previous layers in their structure. Their block training uses local loss functions for HAR applications. Instead of common global backpropagation, local cross-entropy loss and local supervised similarity are used to train each residual block independently. The proposed methodology

improves classification accuracy and memory requirements during the training phase. Cheng et al. [27] propose improving CNN computation using conditionally parameterized convolution, focusing on real-time HAR using mobile and wearable devices.

3.2. Biometric User Identification Using HAR

There are two categories of biometric identification systems. The first is based on static characteristics, including physical features such as the face, fingerprints, and others. The second is based on dynamic features, such as behavioral characteristics, including electrocardiographic signals, voice, or a specific activity. Many studies have explored the possibility of using a smartphone's built-in sensors to recognize user activity. However, few have explored the potential of this data for BUI.

Mekruksavanich et al. [11] propose identifying the user with different physical activity signals, a tri-axial accelerometer, and a tri-axial gyroscope. Its structure recognizes the user's activity and then performs authentication. There is one authentication network for each physical activity in the database. The authors used a CNN and an LSTM to achieve 91.77% and 92.43% accuracy, respectively. The datasets used were the UCI human activity recognition dataset and the USD human activity recognition dataset. The first dataset has data from 30 subjects performing six daily activities. The second dataset includes activity data recorded with 14 subjects, 7 males and 7 females, also performing daily activities. The subjects perform 12 daily activities. The window size was 128 samples with 50 Hz sampling frequency in both datasets.

Juefei-Xu et al. [22] proposed a user identification system based on a previous detection of different walking paces. The walking events are divided into speed categories such as normal and fast. The performance was evaluated in three scenarios. The first, for matching normal versus normal pace, which means that only normal walking samples were considered. The authors achieved a verification rate (VR) of 99.4%, 0.1% for the false acceptance rate (FAR), and 95% for accuracy. In the second scenario, for fast versus fast, they achieved 96.8% for VR, 0.1% for FAR, and 98% for accuracy. In the third, where all samples were used, either normal or fast, the performance dropped to 66.1% for VR, 0.1% for FAR, and 40% for accuracy. FAR measures the average number of false acceptances within a biometric system. This metric is also referred to as the false negative rate. In contrast, VR measures the average number of predicted positive acceptances in the face of ground truth. A tri-axial accelerometer and tri-axial gyroscope are used. The authors used a proprietary database of 36 subjects. Feature extraction by handcrafted methods was employed. The window size used was 1 s. The work showed that the speed at which the user is walking must be considered for correct user identification.

3.3. Gradient-Weighted Class Activation Mapping

Deep networks are usually treated as a black box, where it is impossible to know precisely how the network will behave or why it made a decision. A new branch of research into model explanation called explainable AI (XAI) uses various techniques to explain network decisions. The gradient-weighted class activation mapping (grad-CAM) technique is the most widely used and cited today [28].

Sousa et al. [28] used XAI methods for a COVID-19 classifier on CT images. The authors explain that networks can understand undesired artifacts in databases, and model explanation techniques help avoid this problem. They evaluated some standard methods for XAI systems, such as grad-CAM, LIME, RISE, square grid, and direct gradient approaches. The authors show that spurious artifacts may affect some network architectures. The author reveals that high-performance results can lead to a misconception that they do not suffer from bias problems.

Grad-CAM uses a gradient from a target class, flowing into the final convolutional layer of the architecture, to create a coarse location map that highlights the regions of the image that most influenced the model's decision making [12]. It is known that the deeper convolutional layers can abstract more complex information, so the last convolutional layer

is expected to contain the maximum abstraction of the signal. Visualizing this information gives us a clearer idea of what the intelligence has learned [29]. This visualization becomes more interesting when it is possible to focus only on the information that led the network to associate a specific input with a target class with the help of location maps. Grad-CAM emerged as an improvement on the work of Zhou et al. [19], which proposed class activation mapping (CAM) to identify discriminative regions used by restricted classes for image classification with CNN classifiers. However, the CAM technique faces a limitation since it is only applicable for networks that do not have fully connected layers. Grad-CAM is a generalization of CAM, making it possible to apply it to different families of CNNs, including CNNs with fully connected layers, CNNs with structured outputs, CNNs used in tasks with multimodal inputs, and reinforcement learning.

The grad-CAM technique takes a target signal and a class of interest as input. Then, the gradient is calculated for the target class y^c with regard to the feature maps of the last convolutional layer A^k_{ij}. As described in Equation (1), to obtain each neuron's importance weights $\propto^c_k$, these obtained gradients are passed through a global average pooling function.

$$\alpha^c_k = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A^k_{ij}}}_{\text{gradients via backprop}} \tag{1}$$

The weights $\propto^c_k$ represent a partial linearization of the deep network with respect to A, and capture the "importance" of a feature map for a target class. We then compute a weighted combination of the direct activation maps, followed by a ReLU function to eliminate negative contributions. With this, we get the heatmap of the regions of importance in our network $L^c_{\text{Grad-CAM}}$, according to Equation (2).

$$L^c_{\text{Grad-CAM}} = \text{ReLU} \underbrace{\left(\sum_k \alpha^c_k A^k\right)}_{\text{linear combination}} \tag{2}$$

Grad-CAM is usually applied to images. We will use this technique for a time series of accelerometer data. For this purpose, we have made some modifications to the original method.

We do not need to calculate the global average pooling in two dimensions because it is a time series, only in one dimension, as shown in the following Equation (3):

$$\alpha^c_k = \overbrace{\frac{1}{Z}\sum_i}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A^k_{ij}}}_{\text{gradients via backprop}} \tag{3}$$

For the heatmap, we do not use the ReLU function. We normalize the signal between 0 and 1. When plotting the heatmap, we use the nearest-neighbor interpolation method to interpolate the signal to the same size as the input signal. Linear interpolation is currently the most widely used with grad-CAM. However, when working with signals, the result is not clear enough when we use this method. As seen in Figure 1, if we are interested in seeing well the regions of importance, the result is more evident when we apply the nearest-neighbor interpolation method because the bands are more evident, while in the bilinear interpolation method, these regions of importance seem to be smoother, making it more difficult to distinguish a very important area from one with intermediate importance.

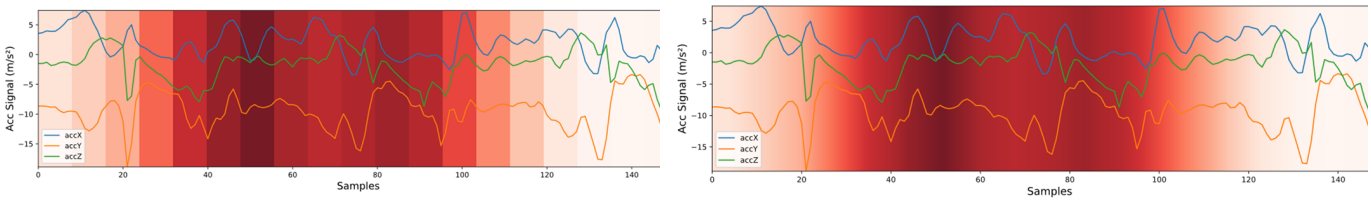


Figure 1. Comparison of nearest-neighbor and bilinear interpolation methods. (**a**) Nearest-neighbor interpolation. (**b**) Bilinear interpolation.

4. Methodology

This section intends to explain the dataset, how the data was trained and divided into each validation strategy, the proposed CNN architectures, and the proposed framework.

4.1. Dataset

The UniMiB-SHAR database is composed mainly of data from women aged between 18 and 60 years, with height between 160 cm and 190 cm, and mass between 50 kg and 82 kg. The device used to acquire the data was a Samsung Galaxy Nexus 19250 with Android operating system 5.1.1, using the Bosh BMA220 acceleration sensor. The sampling frequency was 50 Hz, with a time window of 3 s, resulting in 151 samples for each of the 3 axes of the accelerometer. The windowing process was event-based, based on peak detection. Each activity was performed between 2 and 6 times. Two smartphone positions were considered. Half of the participants placed the smartphone in their left pocket, while the other half placed it in their right pocket [7].

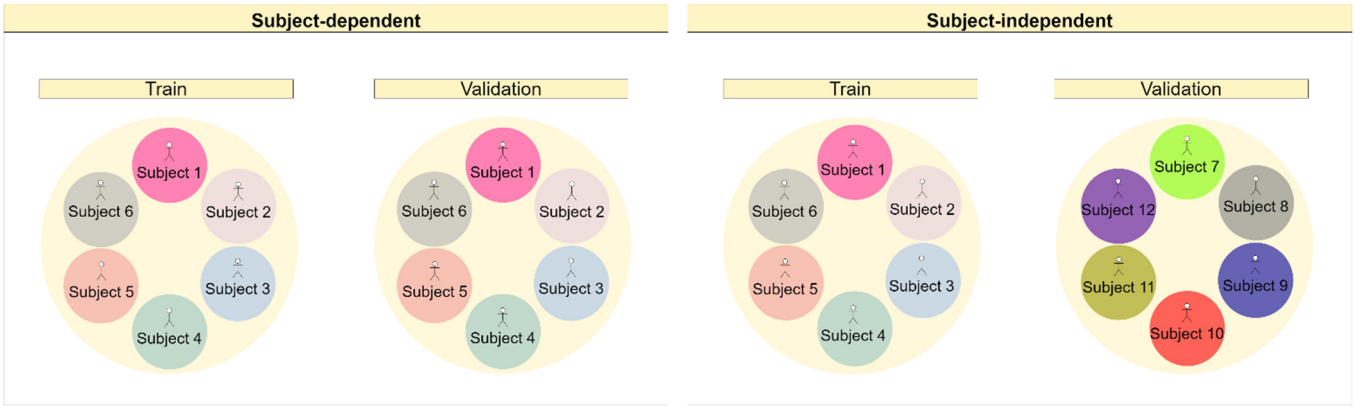
The dataset consists of 11,771 data samples. Thirty subjects performed 17 activities, with 9 types of activities of daily living (ADLs) and 8 types of falls. Table 3 shows more information. As shown in the N° Sample column, the dataset is unbalanced. Finally, as shown in the N° subjects column, not all participants performed all activities.

Table 3. UniMiB-SHAR dataset details.

Action	Label	Activity	N° Samples	N° Subjects
ADL	StandingUpFS	Standing up from Sitting	153	24
	StandingUpFL	Standing up from Laying	216	28
	Walking	Walking	1738	29
	Running	Running	1985	30
	GoingUpS	Going upstairs	921	30
	Jumping	Jumping	746	30
	GoingDownS	Going downstairs	1324	30
	LyingDownFS	Lying down from sitting	296	29
	SittingDown	Sitting Down	200	28
Falls	FallingForw	Falling forward	529	30
	Falling Right	Falling rightward	511	30
	Falling Back	Falling backward	526	30
	Hitting Obstacle	Hitting obstacle	661	30
	Falling With PS	Falling with protection strategies	484	30
	Falling Back SC	Falling backward-sitting-chair	434	30
	Syncope	Syncope	513	30
	Falling Left	Falling leftward	534	30

Splitting the data into the training and validation subsets for activity recognition depends on the chosen validation strategy. For the subject-dependent (SD) strategy, the data were split with shuffle, using the number 42 as a random state seed, using the Sklearn library in Python. The proportion between training and validation was 70/30. This method can also be known as hold-out. For the subject-independent (SI) validation strategy, the related work usually uses 1 subject in the validation. This method is known as leave-one-subject-out. In our work, considering keeping the same proportion of data used in the subject-dependent strategy, to make the comparison of the two approaches equivalent, we chose to use subjects 1 to 9 in validation and the remaining 21 subjects in training, since 9

represents 30% of the total number of individuals. Figure 2 shows a diagram to illustrate the differences between the chosen validation strategies.



(**a**) (**b**)

Figure 2. SD validation strategy and SI validation strategy. (**a**) We train and validate with the same subjects represented in the diagram by using circles with the same colors. In our case, we have 70% of the data for training and 30% for validation, which means that, on average, 70% of the total data from each individual were used in training, and the remaining in validation. (**b**) We have different subjects present in the training and validation, represented by the circles in different colors. In our case, we decided that 30% of the total subjects would be used for validation and the rest for training.

For the biometric user identification, each physical activity was considered a subset. In the case of the walking subset, all walking samples from all individuals in the dataset were filtered to form the walking subset. Then, with this subset, the same split as in the SD strategy was used, with 70/30 training/validation proportion, using the random state seed 42. Figure 3 clarifies the division performed on the data for training the biometric user identification network.

4.2. Architectures

The simulations were performed using 2 convolutional network architectures, CNN1 and CNN2. The architectures were trained and implemented on the TensorFlow 2 framework and had a similar configuration.

Initially, we will expose what is shared by both architectures. In both, the Adam optimizer with default hyperparameters was used. The models were trained for 300 epochs. Each convolutional block was composed of 2 convolutional layers with 100 feature maps each. The ReLU activation function was used. In each max pooling layer, the pool size and step were 2. The value used in the dropout layer is 0.5. The softmax layer has n neurons, representing the number of classes for the particular problem. We used a custom callback to make checkpoints after each training epoch to obtain the best model based on the macro F1-score metric on the validation set. The batch size was 256 samples. Both architectures had received the same signal at their inputs with the shape of 151×3.

Next, we point out the differences between the two architectures. The first architecture has 4 convolutional blocks consisting of 2 convolutional layers, while the second has 3 convolutional blocks. In CNN1, the kernel size was 8, while in CNN2, it was 4. Figure 4 shows both architectures.

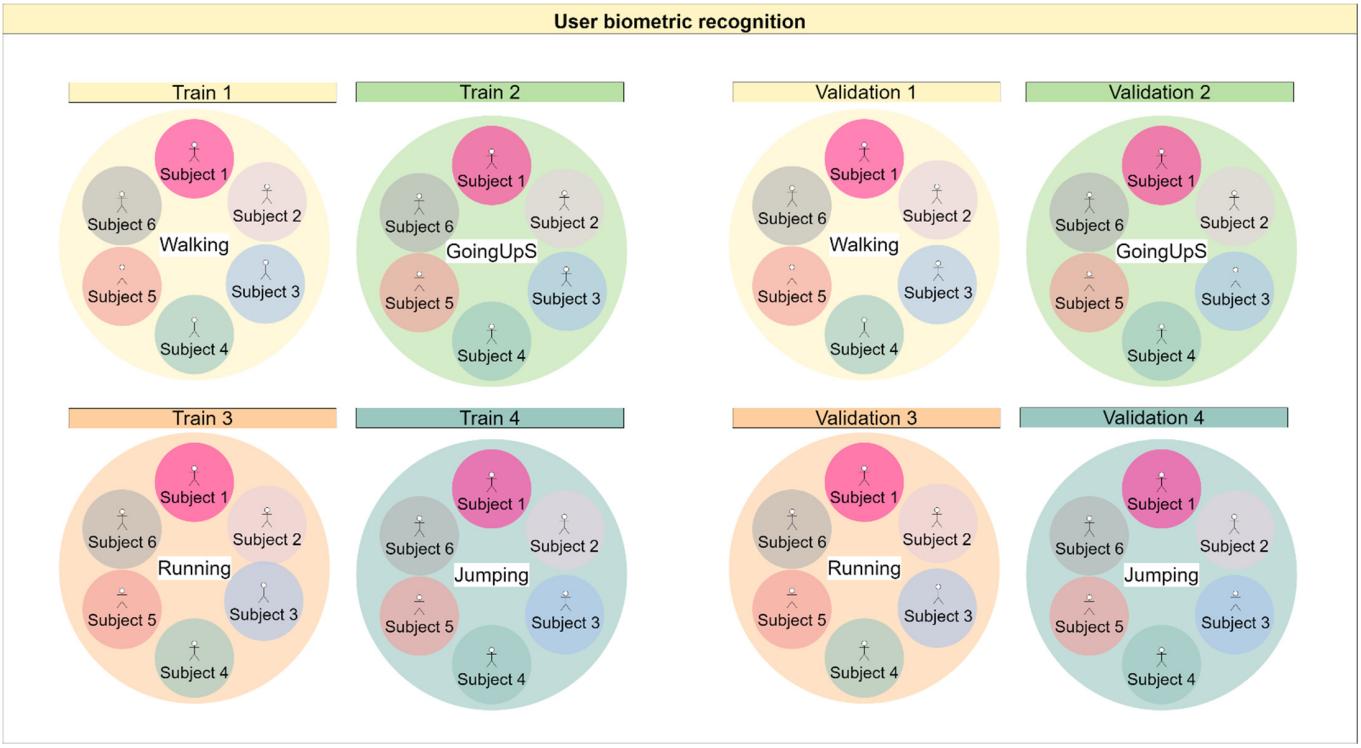
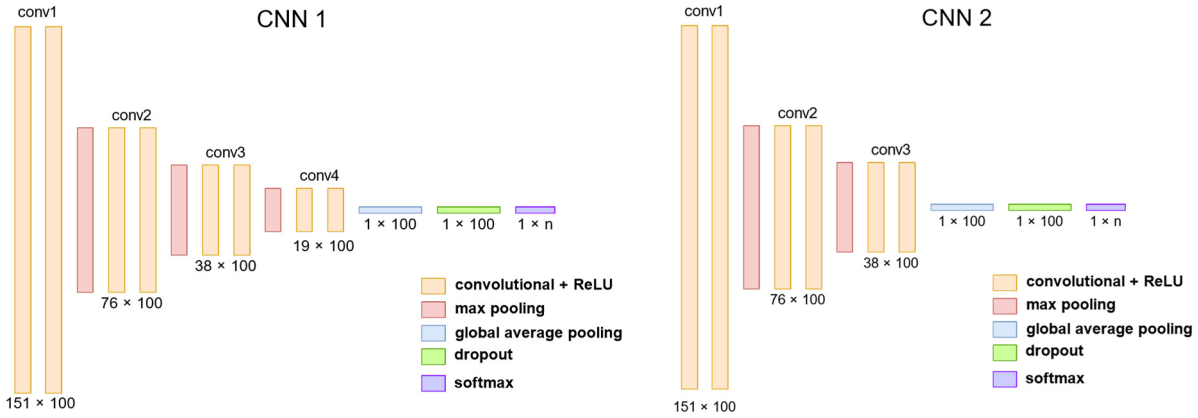


Figure 3. Division of the data for biometric user identification. The larger circles represent separate subsets, and the smaller ones represent different subjects. For example, the running activity is denoted by the orange circle and has a rectangle with the name Train 3 on the left side. There is the same orange circle on the right side, with the name Validation 3, meaning that we have a net trained to recognize individuals who have been trained and validated with just the running activity data.



(**a**) (**b**)

Figure 4. The CNN architectures implemented with TensorFlow 2. (**a**) CNN1 architecture with 4 convolutional blocks. (**b**) CNN2 architecture with 3 convolutional blocks.

4.3. Proposed Framework

To summarize, we propose the framework illustrated in Figure 5. In the database loading step, we performed the database processing and an exploratory analysis of the data. This exploratory analysis allowed us, for example, to identify that not all individuals performed all activities in the UniMiB database. After this, the data was subdivided in the splitting step according to the appropriate validation strategy. In HAR, we had the SD and SI strategies, while in BUI, we had the generation of a subset with physical activity. Then, the models were implemented in the training and evaluation step, and their results were

contrasted using performance metrics. We used the CNN1 and CNN2 architectures for HAR, and for BUI, only the CNN1 architecture.

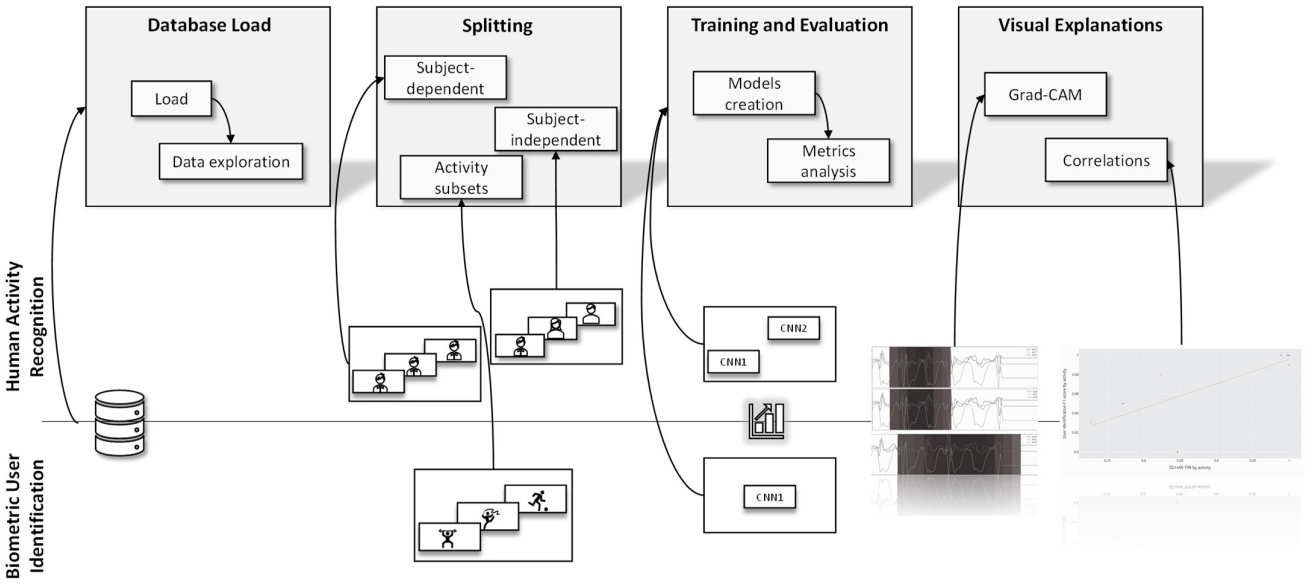


Figure 5. Summary of the framework used for the work. In the top part, we have the processes applied in HAR, in the bottom part, the processes applied in BUI.

The framework is similar to the Activity Recognition Protocol, adding the step of visual explanation. This step allowed a more comprehensive notion of the limitations of the models and database. We generated two graphs, the first with grad-CAM and the second with a correlation between HAR SD and BUI. Using grad-CAM, visual explanations were generated for each class in the dataset, which allowed biases in the database to be identified. Furthermore, these visual explanations gave a clearer idea of the model's potential.

5. Results

In this section, the model results for HAR and BUI will be briefly presented and discussed.

5.1. HAR

We evaluated and trained the CNN1 and CNN2 architectures with the SD and SI strategies. We considered macro average and weighted average metrics to better compare our results with other works. The difference between them is that the former only considers the impact achieved by the metric in each class. In contrast, the latter considers the result of the metric in each class weighted by the number of samples evaluated. The first deals with an unbalanced system better since it does not distinguish between classes with more extensive data. In contrast, the second one provides a more general system idea, as long as the distribution of the classes in the dataset is equal to the distribution of this data in the real world.

Tables 4 and 5 show the performance of each network in more detail, considering the SD and SI strategies, respectively. Appendix A shows the detailed performance of the CNN1 network for the two validation strategies.

Table 4. Performance of CNN1 and CNN2 in HAR, with the SD strategy and 70/30 split.

Model	Macro Average			Weighted Average			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
CNN1	0.9634	0.9624	0.9626	0.9781	0.9779	0.9779	0.9779
CNN2	0.9612	0.9616	0.9610	0.9771	0.9768	0.9768	0.9768

Table 5. Performance of CNN1 and CNN2 in HAR, with the SI strategy and 9 subjects on validation.

Model	Macro Average			Weighted Average			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
CNN1	0.6513	0.6631	0.6530	0.7619	0.7553	0.7565	0.7553
CNN2	0.6361	0.6361	0.6328	0.7321	0.7177	0.7207	0.7177

The results for the SD validation strategy were significantly higher than those achieved with the SI strategy for both trained architectures, considering all metrics. This result was already expected since previous studies point out that the SI validation method is more challenging and is closer to the actual result that a network will have since the network will not have prior knowledge about how individuals perform the activity [4].

In the two scenarios evaluated, CNN1 was superior to CNN2. However, with the SD strategy, both achieved similar results. The difference between the values of macro F1-score obtained by the two architectures differed by only 0.2%. In the SI strategy, it is already possible to see a more significant difference in performance of approximately 2% for the macro F1-score metric and 4% for the accuracy metric.

Although the focus of our article is not to get the best possible architecture in the areas of HAR and BUI, the results achieved are as good as, or even better than, the articles presented in Table 2. Although the validation methods and subsets are different, the model's excellent performance is worth noting. Observing the accuracy metric, our work achieved the same accuracy as the best work with the SD strategy, 0.978, using the CNN1 network. The difference is that this result was achieved with cross-validation. Considering that cross-validation overestimates network performance, the result obtained with the hold-out method is significant [10]. A complete analysis of previous works can be done by looking at Table 2, so we included all our performance metrics obtained with CNN1.

The result for the SI strategy was 0.755 accuracy, which is also close to the baseline. Again, the results achieved in this article are more significant, since the best result was achieved with the LOSO method, which has only one individual in the validation. Ours has nine individuals in the validation set.

The performance metrics show that the best validation strategy was SD, and the best network was CNN1. In Section 6, we will use other evaluation and validation methods to contrast these architectures and choose both the best model and the best validation approach.

Next, we will analyze the diagonals of the confusion matrices shown in Figures 5 and 6. The diagonal is our recall metric, previously defined in Table 1. This metric is also called the true positive rate (TPR). As shown in Figure 6, for CNN1 validated with SD, among the ADLs, the lowest-performing class was StandingUpFL with 90% TPR, followed by LyingDownFS with 93% TPR. The confusion about StandingUpFL was quite reasonable, since 7% of the wrongly predicted samples were mistaken for StandingUpFS, which are indeed similar ADLs. However, among the LyingDownFS confusion, there does not seem to be a logical sense, since most of the confusion, approximately 4%, occurred with the StandingUpFS class. Observing the falling performance of classes, confusion seems to occur between the different types of falling. The lowest-performing class was the FallingRight class with 91% TPR, most frequently confused with the Syncope class.

Figure 7 shows that among the ADLs, the lowest performance for the CNN1 validated with SI came from SittingDown with 44% TPR, followed by LyingDownFS with 66% TPR. The SittingDown confusion shows a possible problem in network learning, where 37% of the samples were confused with GoingDownS, which are entirely different activities. Most of the confusion in the LyingDownFS class, representing 16% of the total, was with the SittingDown class, strengthening the idea that the features learned from the SittingDown class were inappropriate. In Section 6, we will show that the SittingDown and LyingDownFS classes have some bias problems that can affect the network's learning. Among the falling classes, the same pattern identified in SD was found. Most of the confusion

occurs among the falling types, but also with ADL classes, as in FallingBackSC, in which 6% of the samples were confused with LyingDownFS.

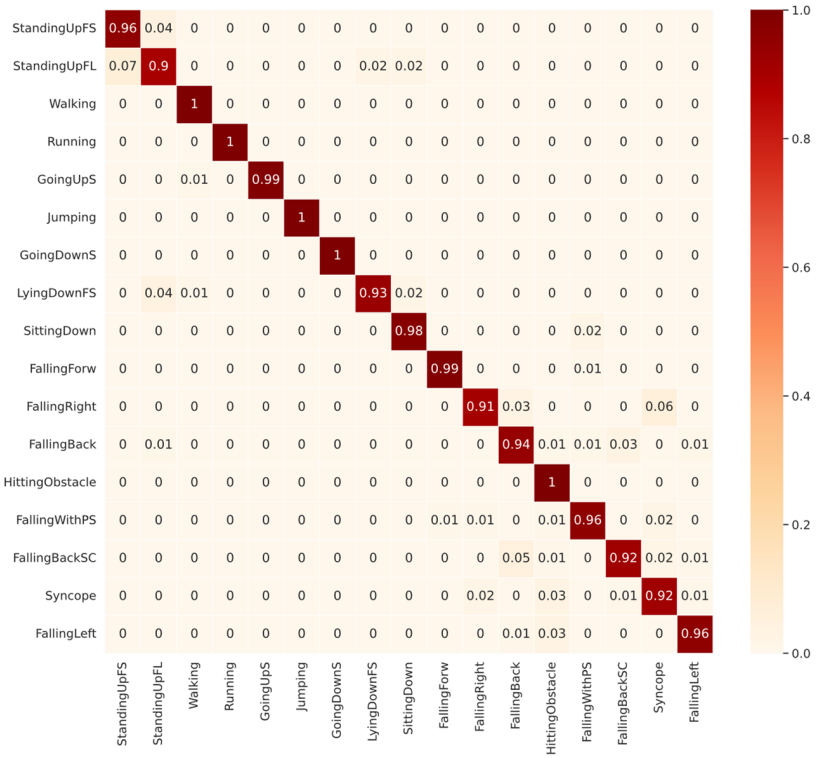


Figure 6. Confusion Matrix of CNN1 with subject-dependent 70/30 for all 17 classes of UniMiB-SHAR dataset.

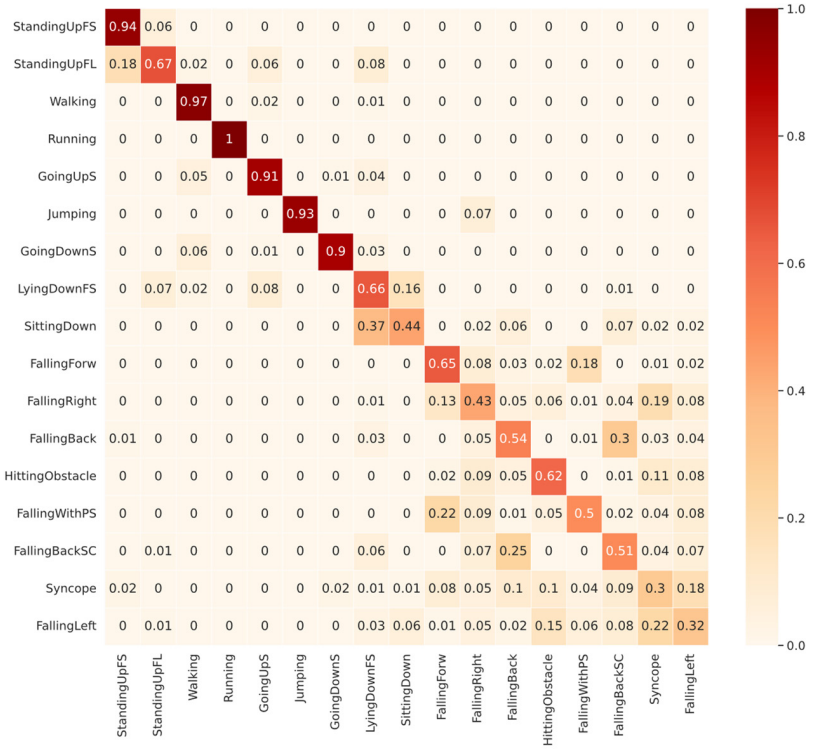


Figure 7. Confusion Matrix of CNN1 with subject-independent with 9 subjects in validation for all 17 classes in UniMiB-SHAR dataset.

In general, in SI, for some classes, the features learned by some subjects were enough to classify the activity performed by the other subjects, such as the Running class. In SD, as the network sees samples from all the people, the classification becomes easier, reflecting higher performances. However, it is not possible to know if these features were relevant enough to perform well for subjects outside the database since we do not have an independent portion of the data with the same distribution to validate the results.

5.2. Biometric User Identification

To prove that the HAR data samples carry a large portion of the user's personal information, we trained and evaluated the CNN1 architecture to perform the subject identification according to the activity developed by the individual. This system in production should be accompanied by a prior classifier that detects the activity performed by the user and only then recognizes the subject, since each identification AI obtained was trained and validated by filtering the database by activity. Using the SD data split shown in Figure 2, we modified only the number of neurons in the last layer of CNN1, softmax, for the number of subjects that performed each activity, according to the column N° subjects in Table 3. Seventeen networks were obtained, one for each activity. Figure 8 shows the performance of each AI developed based on the macro F1-score metric. In contrast, Figure 9 shows the same performance based on the accuracy metric.

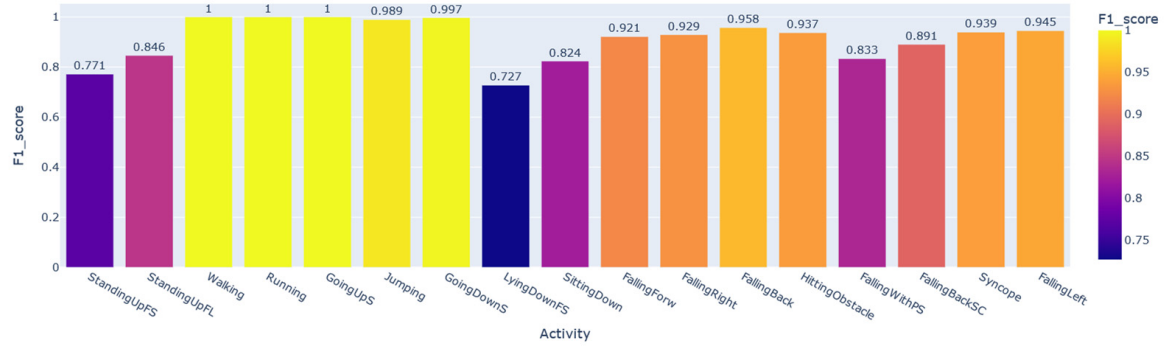


Figure 8. General BUI performance for each activity considering the macro F1-score.

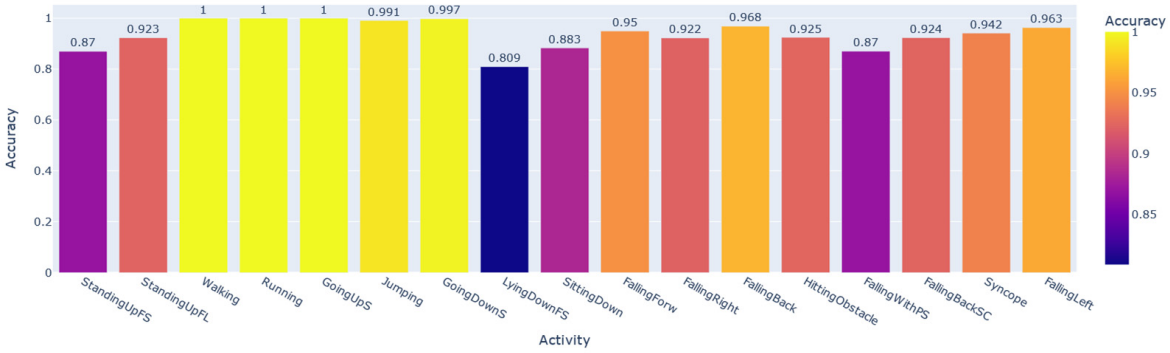


Figure 9. General BUI performance for each activity considering the accuracy.

In some activities, the user's personal information is more significant than in others. For example, in Walking samples, it is possible to recognize each of the individuals with a 100% macro F1-score. At the same time, for the LyingDownFS subset, this value drops to 73%. Among the most common everyday physical activities that achieved the highest performances were Walking, Running, GoingUpS, and GoingDownS.

Later in the discussion section, we will associate the performance in recognizing the activity with the performance in identifying the subject.

Overall, we can see that it is possible to recognize the individuals well, with more than 91% average of all macro F1-scores obtained for each subset. Considering the accuracy, this

percentage rises to 93%. In the next section, we will demonstrate that the StandingUpFS, StandingUpFL, LyingdownFS, and SittingDown classes are activities that suffer from database bias problems. These were precisely the lowest-performing subsets of our AI for user identification, showing that the performance of this network for identification would be even higher if we disregarded the performance for these activities.

6. Discussion

In this section, we explain the results obtained by each trained network, either for HAR or BUI. We explain possible reasons for the difference in performance between the SI and SD strategies using analyses involving the identification networks. We then use the visual explanations of the grad-CAM technique to visualize what the networks took into account for decision making. Finally, with the same technique, we find inconsistencies in our database and show how these inconsistencies affected learning in the networks.

6.1. Explaining the HAR and BUI Correlationship

Table 3 shows that not all subjects performed all activities in the UniMiB-SHAR dataset. Furthermore, in Table 2, many papers used the SI approach with a leave-one-subject-out validation strategy. This validation approach is not the appropriate strategy for this scenario, since most machine learning models need independent and identically distributed data (IID) [30]. Since not all individuals performed all activities in this dataset, using this strategy does not meet the criteria of having IID data.

Furthermore, many papers claim to have surpassed the performance baseline and to be the current state of the art of the UniMiB-SHAR dataset [20,24]. However, some works do not use the same validation strategy as their baseline, making the results noncomparable. Some papers use the same validation strategy, but with different subsets of data, by not using an equivalent data split [21,25]. For example, most data splits are done randomly, but every split must be based on the same random seed to ensure the reproducibility of the results. However, some authors do not pay attention to this fact, obtaining no equivalent subsets, which implies that, in some cases, we may have similar samples in the training and validation sets, making the performance high. In contrast, there may be fewer similar samples in the training and validation subsets in other cases, raising the problem of generalization of the network, probably resulting in lower performance.

In the previous section, it was shown that the nets recognize all 17 activities well, but there is a difference of over 31% between the two validation strategies for the macro F1-score metric. We can also see a high performance in user identification per activity. Figure 10 shows a combination of these performances.

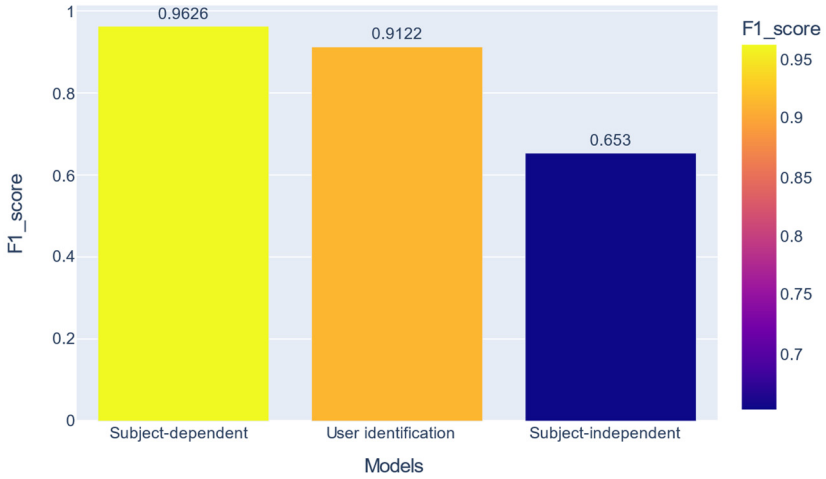


Figure 10. Comparison between the best performances obtained with HAR with SD, biometric user identification, and HAR with SI.

Previous work has mentioned that all activity data contain personal user information [31]. Our work shows that the user signature is more potent in some activities than others. For example, in the Walking activity, we can identify the user with 100% F1-score macro, while in the StandingUpFS activity, this metric drops to 77.1%.

Figure 11 shows a correlation between the recognition performance of ADLs with the SD strategy, as well as the user identification performance for these ADLs. The Pearson coefficient and the p-value for testing noncorrelation are 0.775 and 0.014, respectively. Pearson's coefficient measures the linear relationship between two distributions. Correlations of -1 and $+1$ imply a perfect positive correlation and a perfect negative correlation. The p-value exposes the probability of these correlations occurring at random.

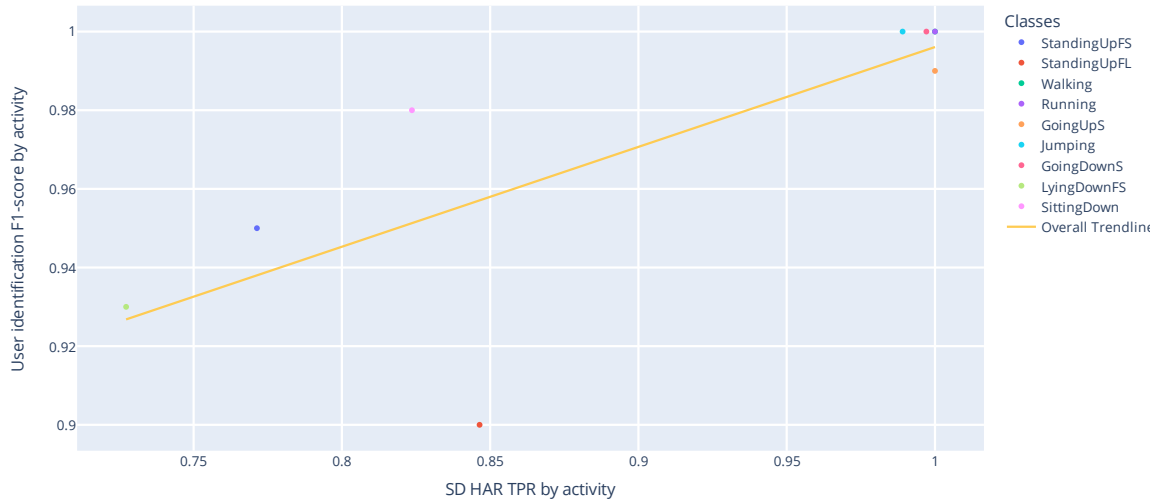


Figure 11. Correlation between the performance of user identification and SD HAR for ADLs.

The strong correlation obtained for ADLs may explain the difference between the SD and SI validation strategies. Basically, with the SD strategy, the network learns the activities' patterns and personal signatures from the subjects. The result is that the network may memorize how all individuals do the activities and does not learn the correct patterns to recognize the activity.

As shown in Figure 12, considering ADLs and Falls, the Pearson correlation and p-value obtained were 0.504 and 0.039. When we consider falls, the correlation is weaker. However, falls are not commonly used to recognize the user in the same way as ADLs.

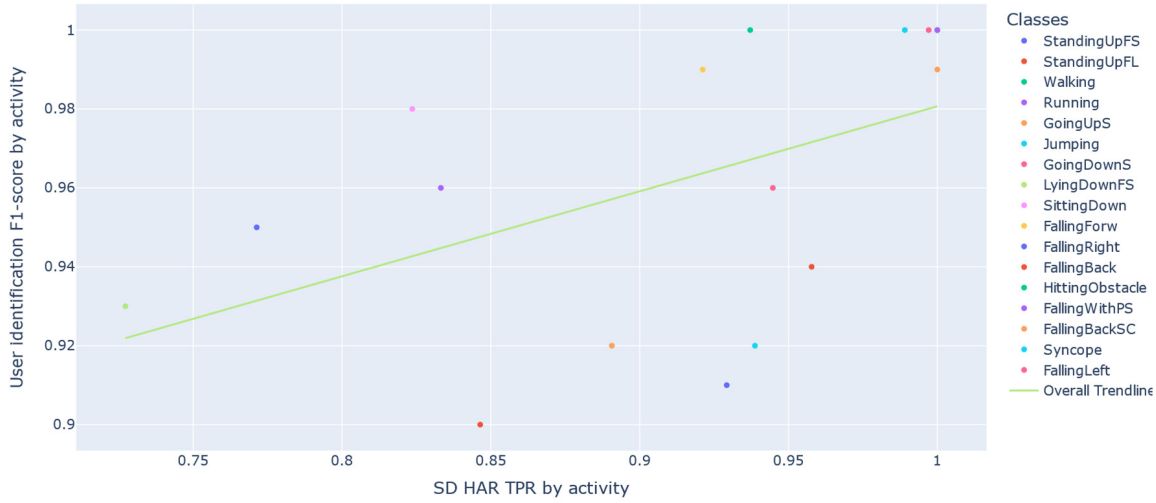


Figure 12. Correlation between the performance of user identification and SD HAR.

In general, in each activity analyzed, the performance of the classifier trained with SD may not indicate how well this classifier will recognize the activities in real-life scenarios.

6.2. Interpreting CNN HAR Models with Gradient Class Activation Mapping

Interpreting the models' predictions is essential to evaluate their generalization capability and understand their limitations. Grad-CAM provides a way to perform this analysis based on visual information about the regions of the input signal that most influence the model's decision making. Figure 13 shows what the CNN1 architecture has learned and considered to predict Walking activity.

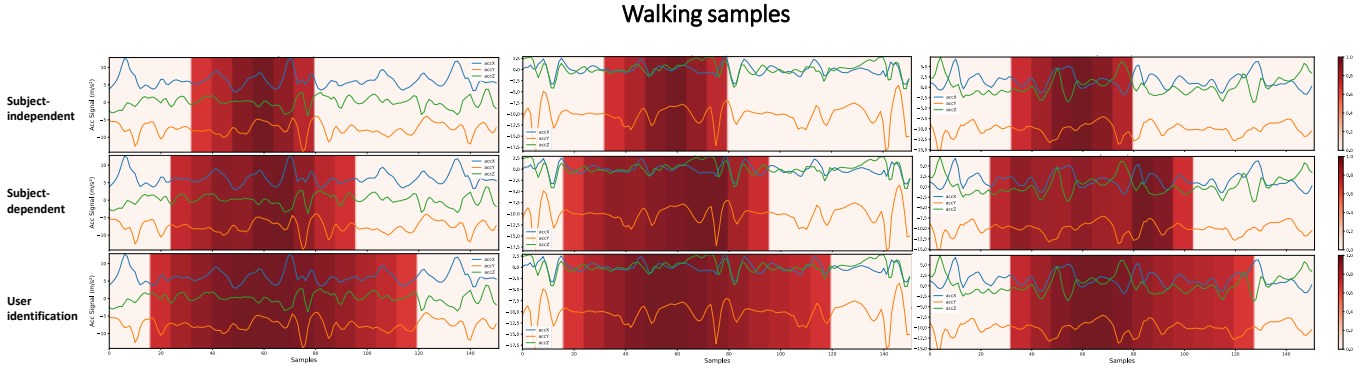


Figure 13. Higher-importance segments for decision making in the CNN1 architecture, for HAR with SI, HAR with SD, and user identification, considering Walking samples with different subjects. The darker the area, the more critical it is for decision making. To obtain these heatmaps, a threshold of 0.7 was considered in the grad-CAM result in order to make more evident the regions of greater importance.

There are key patterns that help identify each activity. A network can learn this key event or other hidden patterns to identify an activity. However, the larger the region that the architecture considers for decision making, the less likely it is to have learned only these key patterns. We are aiming to demonstrate that when the network, in decision making, considers a vast region of the signal, there may be a portion of the individuals' personal information that was learned during training. A user identification network must consider a larger part of the signal, since it must learn the person's digital signature in addition to the key events of the activities. Figure 13 shows that the SI, SD, and user identification networks gradually consider larger portions of the signal to make predictions. It is probable that the more extensive the region is taken into account in decision making, the more personal information from the user the network is learning.

Some activities have multiple key events in a single sample. For example, in the Running activity, where key events are associated with steps, in the 3 s period of a sample, the number of key events is greater than in the Walking activity. Since CNN1 has only 19 $\times$ 1 resolution in its last layer, if our class has many key events, the network for decision making considers more extensive regions of the signal, as shown in Figure 14. Thus, for this activity, the HAR networks with SI and HAR with SD obtained closer results than in Figure 13, not necessarily because the network learned the user's characteristics. In this case, the network learned an activity with multiple key events. Furthermore, it is still possible to see that the HAR with SI networks, for decision making, generally takes into account a less extensive region of the signal. At the same time, the authentication network continued to show the same pattern as the previous activity, considering a large region of the signal as important.

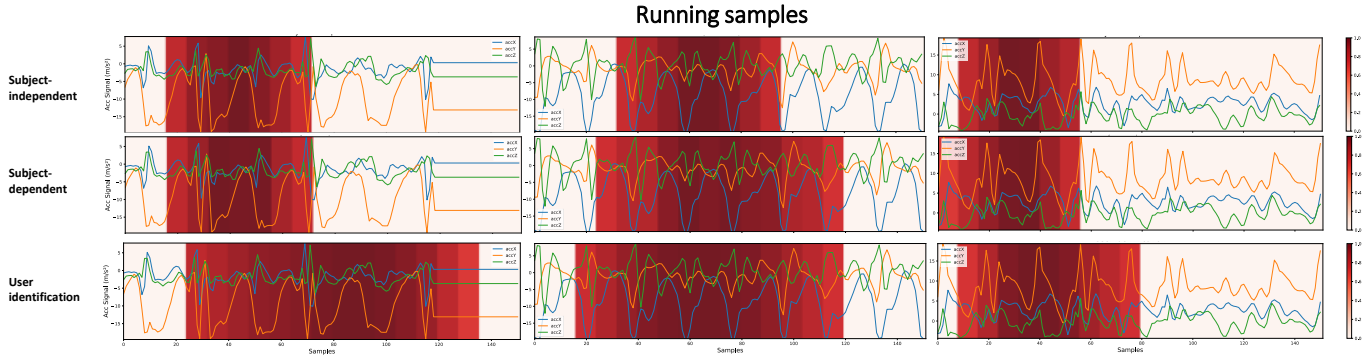


Figure 14. The most critical decision-making segments in the CNN1 architecture for HAR with SI, HAR with SD, and user identification, considering samples of the Running class with different subjects. To obtain the heatmaps, a threshold of 0.7 was considered in the grad-CAM result in order to make the most important regions more evident.

As shown in Figure 3, the CNN2 architecture, if compared to the CNN1 architecture, has a higher resolution (number of neurons) in the last convolutional layer. While CNN1 has 19 $\times$ 1, CNN2 has 38 $\times$ 1 resolution. Figure 15 shows that this higher resolution allows the network to learn key events more precisely. We contrast the learning between the SI and SD approaches through this figure. We remove the visualizations from the user identification networks since it always considers a large signal region, as shown in Figures 13 and 14. We can see that the SI and SD strategies learn key events, but the SD network also gives secondary importance to the signal as a whole. The user's signature is what the network is learning among the key events. The influence this may have on the final result is that the network may memorize how users in the database perform activities rather than learning the specific physical activity patterns. The portions of secondary importance between the key events in the SI network are less spread out than in the SD network.

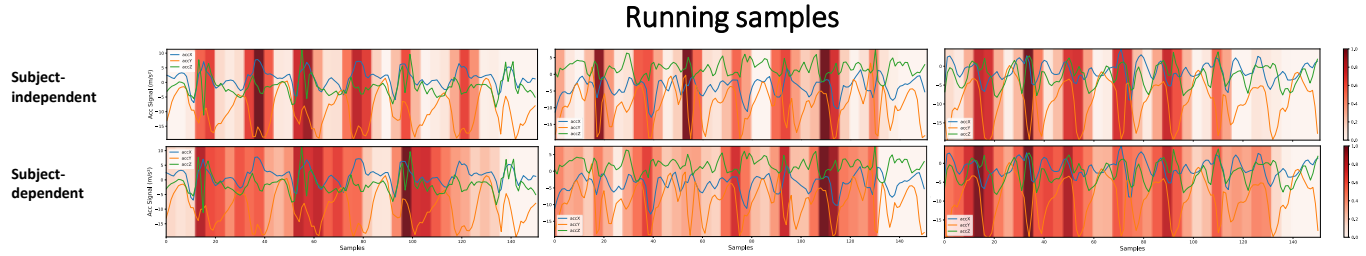


Figure 15. Higher-importance segments for decision making in the CNN2 architecture, for HAR with SI and HAR with SD, considering samples of the Running class with different subjects. To obtain the heatmaps, no threshold was considered in the grad-CAM result in order to make more transparent the smaller, but significant, importance zones that the SD strategy takes into account.

Next, as shown in Figure 16, the learning of the CNN1 and CNN2 architectures is compared using the two validation strategies.

It can be seen that, often, the CNN2 network for decision making takes into account a smaller signal region. Having a smaller region does not always mean we have a better model. For example, the CNN1 network learned more complex features about the signal when, for decision making, it took into account a more extensive region of the signal. At the same time, the CNN2 network learned fewer complex events that can more easily occur in other samples outside the database, leading to incorrect predictions.

A model may be very robust when evaluated through its performance metrics. However, it may not be able to generalize well in real-world scenarios. As shown in Figure 16, although the architectures CNN1 and CNN2 trained with the SD strategy learn different features, their performance metrics differed by only 0.2%. The CNN1 network can learn more complex features than the CNN2 network. We also conclude that the CNN1 network

appears more robust in that it does not give importance to key events as short as CNN2, making this network less susceptible to noise. This analysis would not be possible without the grad-CAM technique's visual exploration.

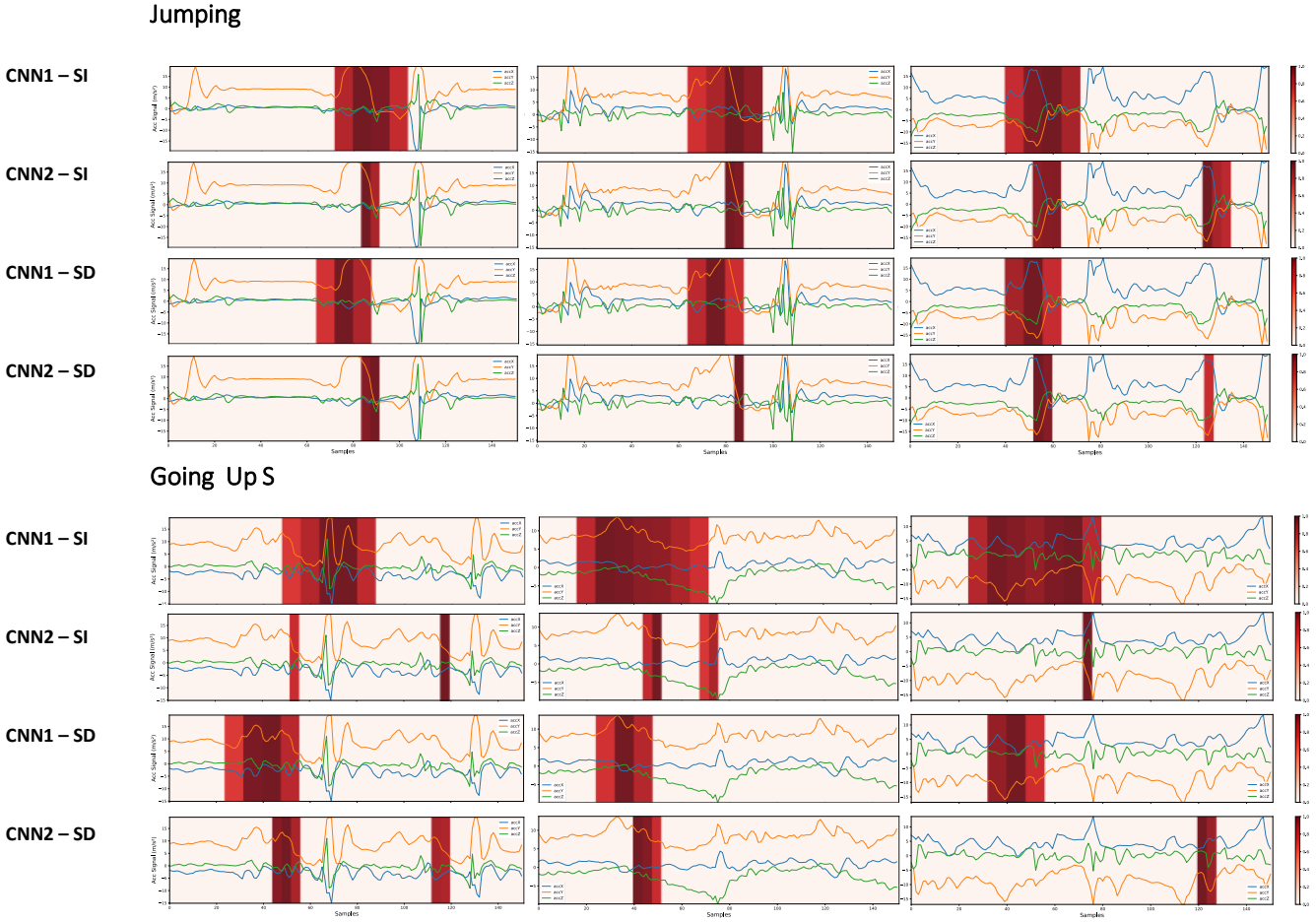


Figure 16. Most important portions for decision making in the CNN1 and CNN2 networks with the SD and SI strategies, for samples of the Jumping and GoingUpS activities. To obtain the heatmaps, a threshold of 0.7 was considered in the grad-CAM result to make the critical regions more evident.

6.3. Identifying Dataset Bias Problems

A machine learning model can learn from data and is, therefore, called data-driven. However, if the data has some problem or some bias, the model inevitably learns this noisy information.

Analyzing the dataset is an important task, but it becomes challenging with extensive databases. Some problems are easy to identify, such as missing data. However, there can also be more complex situations that are difficult to map. Visual exploration through grad-CAM can also find bias problems in the database.

Selvaraju et al. [12] used these techniques to find a problem in their database. The authors were developing an AI model to recognize doctors and nurses, regardless of the gender of the person. The model took an image as input and made a binary classification between the two possible classes. Evaluating the model on its dataset, the performance metrics showed that the model generalized well. However, the result was quite different when testing images outside the dataset. So, they decided to apply grad-CAM to help solve the problem. Analyzing the regions of importance for predicting the images, the authors observed that the model looked at the person's face and hairstyle to distinguish between doctors and nurses. This explained why the model classified most men as doctors and most women as nurses. The model learned a gender stereotype in its database: mostly male

doctors and mostly female nurses. After this analysis, the authors added more images of female doctors and male nurses, solving the problem.

In this work, we used the grad-CAM visual explanations to identify possible bias issues in the UniMiB-SHAR database. Analyzing sample predictions from the CNN2 architecture, we found some strange events in the StandingUpFS, StandingUpFL, LyingDownFS, and SittingDown classes, as shown in Figure 17. This same pattern was repeated in both the SD and SI approaches for the CNN2 architecture. However, since the SD approach achieved higher performance metrics, we chose it to evaluate the results.

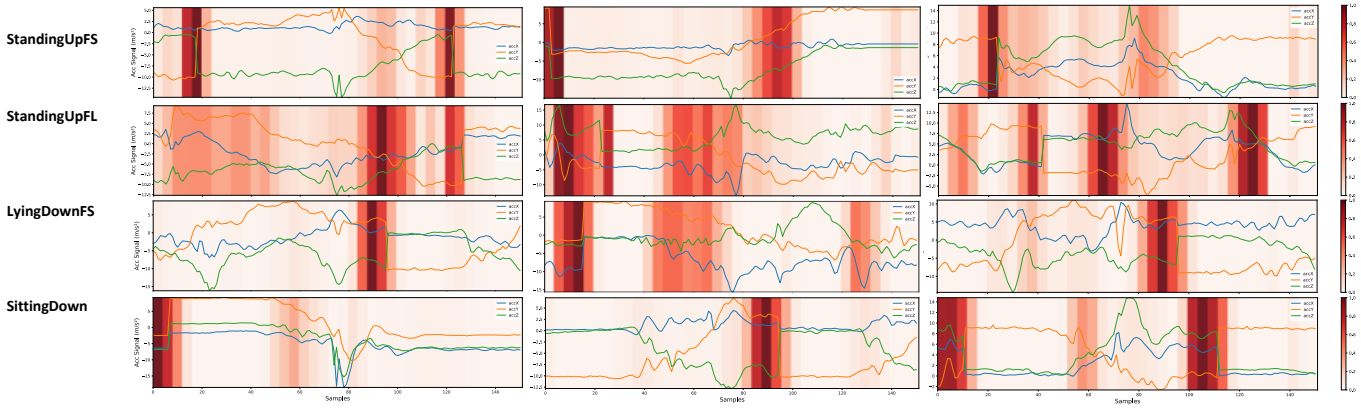


Figure 17. Possible bias problems in the dataset were identified from visual analysis of the grad-CAM in the samples. No threshold was considered for the heatmaps. We can see that the network gave too much importance to discontinuities when predicting. This will probably reflect a problem in the generalization of the model.

Taking the SittingDown class as an example, the network concentrated its prediction on noisy segments. For the first sample of this class, we have an abrupt discontinuity on the y-axis and z-axis. For example, in Figure 17, for the first StandingUpFS sample, in the upper left corner between segments 18 and 19, the signal rises from -10 to 1.3 on the y-axis and falls from 2.8 to -9.2 on the z-axis. Although this sample is from the sitting activity, the user cannot make such a rapid movement. Considering the sampling frequency of 50 Hz, the user would have to perform this movement in 0.02 s. These sections of discontinuities appear in some samples of the three classes mentioned. The discontinuities can occur for several reasons. It can be a problem in the sensor, a problem in the application collecting the data, or even a problem in the windowing process for these samples. Sensor problems can be due to poor calibration. Problems in the collection application may cause the sampling frequency to not be constant, or there may be time lapses due to a crash in the collection application.

Regarding the windowing problems, this may have occurred when assembling the dataset, where a section of a sample may have been merged with the subsequent sample. For example, for each activity, six trials were performed. The windowing process may have caused samples from one trial to be merged with segments from the other trial. Based on our observation, these discontinuities may occur in all three axes or only in the y- and z-axis.

We noticed that the model did not learn the important events to detect the activity, it only focused on the parts where these noises occur. Our network has learned to look at discontinuities to differentiate samples from these four classes. In a real scenario, the network will not find such a pattern, and the result tends to be bad.

In general, both architectures were influenced by the discontinuities present in the dataset. As expected, the CNN2 network was more affected as it generally focuses on smaller signal regions.

Discontinuity problems were also found in the fall classes, affecting the network performance. Analyzing the SD and SI confusion matrices presented in Figures 6 and 7, it was observed that the classes with the highest performance difference had a bias problem in

the dataset. Among the ADLs, the classes with the most significant performance differences were SittingDown, LyingDownFS, and StandingUpFL, with 54%, 27%, and 23% differences in SD and SI performance, respectively, considering the TPR metric. Bias problems were identified in the three mentioned classes, as shown in Figure 17. The performance in the fall classes differed among all classes, but the same issues of discontinuities were identified. Thus, we can conclude that if there is a high difference, in a target class, between the performances of HAR networks in SI and SD, the dataset probably suffers from a bias problem in that class.

Author Contributions: Conceptualization, G.A., C.F.F.C.F. and M.G.F.C.; methodology, G.A., C.F.F.C.F. and M.G.F.C.; software, G.A.; validation, G.A.; formal analysis, G.A., C.F.F.C.F. and M.G.F.C.; investigation, G.A.; resources, C.F.F.C.F. and M.G.F.C.; data curation, G.A.; writing—original draft preparation, G.A.; writing—review and editing, G.A., C.F.F.C.F. and M.G.F.C.; visualization, G.A.; supervision, C.F.F.C.F. and M.G.F.C.; project administration, C.F.F.C.F. and M.G.F.C.; funding acquisition, M.G.F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 48 of Decree n°6.008/2006 (SU-FRAMA), and was funded by Samsung Electronics of Amazonia Ltd., under the terms of Federal Law n°8.387/1991 through agreement 001/2020, signed with UFAM and FAEPI, Brazil. The authors would also like to thank CAPES for its financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The UniMiB-SHAR dataset can be found on http://www.sal.disco.unimib.it/technologies/unimib-shar/ (accessed on 23 April 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. CNN1—Subject-dependent detailed performance macro average.

Class	Precision	Recall	F1-Score	Support
StandingUpFS	0.9216	0.9592	0.9400	49
StandingUpFL	0.9000	0.9000	0.9000	60
Walking	0.9943	1	0.9971	523
Running	1	0.9984	0.9992	614
GoingUpS	1	0.9928	0.9964	276
Jumping	1	1	1	228
GoingDownS	0.9975	1	0.9987	393
LyingDownFS	0.9875	0.9294	0.9576	85
SittingDown	0.9516	0.9233	0.9672	60
FallingForw	0.9935	0.9935	0.9935	155
FallingRight	0.9653	0.9085	0.9360	153
FallingBack	0.9119	0.9416	0.9265	154
HittingObstacle	0.9384	1	0.9682	198
FallingWithPS	0.9762	0.9609	0.9685	128
FallingBackSC	0.9449	0.9160	0.9302	131
Syncope	0.9193	0.9193	0.9391	161
FallingLeft	0.9753	0.9573	0.9662	164

Table A2. CNN2—Subject-dependent detailed performance macro average.

Class	Precision	Recall	F1-Score	Support
StandingUpFS	0.9787	0.9388	0.9583	49
StandingUpFL	0.9219	0.9833	0.9516	60
Walking	1	1	1	523
Running	1	0.9984	0.9992	614
GoingUpS	1	0.9964	0.9982	276
Jumping	1	1	1	228
GoingDownS	0.9949	1	0.9975	393
LyingDownFS	0.9868	0.8824	0.9317	85
SittingDown	0.9077	0.9833	0.9440	60
FallingForw	0.9730	0.9290	0.9505	155
FallingRight	0.9097	0.9216	0.9156	153
FallingBack	0.9536	0.9351	0.9443	154
HittingObstacle	1	0.9949	0.9975	198
FallingWithPS	0.9323	0.9688	0.9502	128
FallingBackSC	0.9084	0.9084	0.9084	131
Syncope	0.9375	0.9317	0.9346	161
FallingLeft	0.9357	0.9756	0.9552	164

References

1. Zhang, S.; Li, Y.; Zhang, S.; Shahabi, F.; Xia, S.; Deng, Y.; Alshurafa, N. Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances. *Sensors* **2022**, *22*, 1476. [CrossRef] [PubMed]
2. Meticulous Research. Wearable Devices Market by Product Type (Smartwatch, Earwear, Eyewear, and others), End-Use Industry(Consumer Electronics, Healthcare, Enterprise and Industrial, Media and Entertainment), Connectivity Medium, and Region—Global Forecast to 2025. Available online: https://www.meticulousresearch.com/product/wearable-devices-market-5050 (accessed on 24 July 2022).
3. Booth, F.W.; Roberts, C.K.; Laye, M.J. Lack of exercise is a major cause of chronic diseases. *Compr. Physiol.* **2012**, *2*, 1143–1211. [CrossRef] [PubMed]
4. Ferrari, A.; Micucci, D.; Mobilio, M.; Napoletano, P. Trends in human activity recognition using smartphones. *J. Reliab. Intell. Environ.* **2021**, *7*, 189–213. [CrossRef]
5. Medrano, C.; Igual, R.; Plaza, I.; Castro, M. Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS ONE* **2014**, *9*, e94811. [CrossRef] [PubMed]
6. Serrão, M.K.; Aquino, G.D.A.E.; Costa, M.G.F.; Filho, C.F.F.C. Human Activity Recognition from Accelerometer with Convolutional and Recurrent Neural Networks. *Polytechnica* **2021**, *4*, 15–25. [CrossRef]
7. Micucci, D.; Mobilio, M.; Napoletano, P. UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones. *Appl. Sci.* **2017**, *7*, 1101. [CrossRef]
8. Godfrey, A.; Hetherington, V.; Shum, H.; Bonato, P.; Lovell, N.; Stuart, S. From A to Z: Wearable technology explained. *Maturitas* **2018**, *113*, 40–47. [CrossRef] [PubMed]
9. Huang, Q.; Hao, K. Development of cnn-based visual recognition air conditioner for smart buildings. *J. Inf. Technol. Constr.* **2020**, *25*, 361–373. [CrossRef]
10. Bragança, H.; Colonna, J.G.; Oliveira, H.A.B.F.; Souto, E. How Validation Methodology Influences Human Activity Recognition Mobile Systems. *Sensors* **2022**, *22*, 2360. [CrossRef] [PubMed]
11. Mekruksavanich, S.; Jitpattanakul, A. Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models. *Electronics* **2021**, *10*, 308. [CrossRef]
12. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626. [CrossRef]
13. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, J.; Havinga, P.J. A survey of online activity recognition using mobile phones. *Sensors* **2015**, *15*, 2059–2085. [CrossRef] [PubMed]

14. Vavoulas, G.; Chatzaki, C.; Malliotakis, T.; Pediaditis, M.; Tsiknakis, M. The MobiAct Dataset: Recognition of Activities of Daily Living using Smartphones. In Proceedings of the ICT4AgeingWell, Rome, Italy, 21–22 April 2016; pp. 143–151.
15. Sadiq, S.; Massie, S.; Wiratunga, N.; Cooper, K. Learning, Deep and Shallow Features for Human Activity Recognition. In Proceedings of the Knowledge Science, Engineering and Management, Melbourne, VIC, Australia, 19–20 August 2017; pp. 469–482.
16. Su, X.; Tong, H.; Ji, P. Activity Recognition with Smartphone Sensors. *Tsinghua Sci. Technol.* **2014**, *19*, 235–249.
17. Rad, N.M.; van Laarhoven, T.; Furlanello, C.; Marchiori, E. Novelty detection using deep normative modeling for imu-based abnormal movement monitoring in parkinson's disease and autism spectrum disorders. *Sensors* **2018**, *18*, 3533. [CrossRef]
18. Ferrari, A.; Micucci, D.; Mobilio, M.; Napoletano, P. On the Personalization of Classification Models for Human Activity Recognition. *IEEE Access* **2020**, *8*, 32066–32079. [CrossRef]
19. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. Available online: http://cnnlocalization.csail.mit.edu (accessed on 24 July 2022).
20. Li, F.; Shirahama, K.; Nisar, M.A.; Köping, L.; Grzegorzek, M. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors* **2018**, *18*, 679. [CrossRef] [PubMed]
21. Teng, Q.; Zhang, L.; Tang, Y.; Song, S.; Wang, X.; He, J. Block-Wise Training Residual Networks on Multi-Channel Time Series for Human Activity Recognition. *IEEE Sens. J.* **2021**, *21*, 18063–18074. [CrossRef]
22. Juefei-Xu, F.; Bhagavatula, C.; Jaech, A.; Prasad, U.; Savvides, M. Gait-ID on the Move: Pace Independent Human Identification Using Cell Phone Accelerometer Dynamics. In Proceedings of the 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, USA, 23–27 September 2012.
23. Ronao, C.A.; Cho, S.-B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.* **2016**, *59*, 235–244. [CrossRef]
24. Mukherjee, D.; Mondal, R.; Singh, P.K.; Sarkar, R.; Bhattacharjee, D. EnsemConvNet: A deep learning approach for human activity recognition using smartphone sensors for healthcare applications. *Multimed. Tools Appl.* **2020**, *79*, 31663–31690. [CrossRef]
25. Tang, Y.; Teng, Q.; Zhang, L.; Min, F.; He, J. Layer-Wise Training Convolutional Neural Networks with Smaller Filters for Human Activity Recognition Using Wearable Sensors. *IEEE Sens. J.* **2020**, *21*, 581–592. [CrossRef]
26. Lv, T.; Wang, X.; Jin, L.; Xiao, Y.; Song, M. A Hybrid Network Based on Dense Connection and Weighted Feature Aggregation for Human Activity Recognition. *IEEE Access* **2020**, *8*, 68320–68332. [CrossRef]
27. Cheng, X.; Zhang, L.; Tang, Y.; Liu, Y.; Wu, H.; He, J. Real-time Human Activity Recognition Using Conditionally Parametrized Convolutions on Mobile and Wearable Devices. *IEEE Sens. J.* **2022**, *22*, 5889–5901. [CrossRef]
28. de Sousa, I.P.; Vellasco, M.M.B.R.; da Silva, E.C. Explainable artificial intelligence for bias detection in covid ct-scan classifiers. *Sensors* **2021**, *21*, 5657. [CrossRef] [PubMed]
29. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. Available online: http://www.image-net.org/challenges/LSVRC/2012/results.html (accessed on 24 July 2022).
30. Wong, T.-T. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognit.* **2015**, *48*, 2839–2846. [CrossRef]
31. Shen, C.; Chen, Y.; Yang, G. On motion-sensor behavior analysis for human-activity recognition via smartphones. In Proceedings of the 2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA), Sendai, Japan, 29 February–2 March 2016; pp. 1–6. [CrossRef]

Artigo 2 - Explaining and Visualizing Embeddings of One-Dimensional Convolutional Models in Human Activity Recognition Tasks

sensors

MDPI

Article

Explaining and Visualizing Embeddings of One-Dimensional Convolutional Models in Human Activity Recognition Tasks

Gustavo Aquino, Marly Guimarães Fernandes Costa and Cícero Ferreira Fernandes Costa Filho *

R&D Center in Electronic and Information Technology, Federal University of Amazonas, Manaus 69077-000, Brazil
* Correspondence: ccosta@ufam.edu.br

Abstract: Human Activity Recognition (HAR) is a complex problem in deep learning, and One-Dimensional Convolutional Neural Networks (1D CNNs) have emerged as a popular approach for addressing it. These networks efficiently learn features from data that can be utilized to classify human activities with high performance. However, understanding and explaining the features learned by these networks remains a challenge. This paper presents a novel eXplainable Artificial Intelligence (XAI) method for generating visual explanations of features learned by one-dimensional CNNs in its training process, utilizing t-Distributed Stochastic Neighbor Embedding (t-SNE). By applying this method, we provide insights into the decision-making process through visualizing the information obtained from the model's deepest layer before classification. Our results demonstrate that the learned features from one dataset can be applied to differentiate human activities in other datasets. Our trained networks achieved high performance on two public databases, with 0.98 accuracy on the SHO dataset and 0.93 accuracy on the HAPT dataset. The visualization method proposed in this work offers a powerful means to detect bias issues or explain incorrect predictions. This work introduces a new type of XAI application, enhancing the reliability and practicality of CNN models in real-world scenarios.

Keywords: human activity recognition; accelerometer data; deep learning; one-dimensional convolutional neural networks; embeddings; explainable artificial intelligence; embeddings visualization; t-SNE; visualization

Citation: Aquino, G.; Costa, M.G.F.; Filho, C.F.F.C. Explaining and Visualizing Embeddings of One-Dimensional Convolutional Models in Human Activity Recognition Tasks. *Sensors* **2023**, *23*, 4409. https://doi.org/10.3390/s23094409

Academic Editor: Joel J. P. C. Rodrigues

Received: 28 February 2023
Revised: 7 April 2023
Accepted: 27 April 2023
Published: 30 April 2023

1. Introduction

Artificial Intelligence (AI) refers to the ability of machines to perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation. AI forms the basis of any technology used for Human Activity Recognition (HAR) [1,2]. HAR systems identify human activities using data captured by wearable devices or smartphones [3,4]. The main goal of HAR is to enhance health outcomes for people who are suffering from chronic diseases, such as diabetes, and Parkinson's disease, and even for older adults [1,5,6]. HAR has numerous applications in the healthcare and wellness sector for individuals who wish to monitor their health and fitness [3,4].

A variety of sensors can be utilized to implement HAR. One compelling sensor in this domain is the accelerometer. The accelerometer can detect high- and low-frequency movements [3,7]. A gyroscope and a magnetometer can be used to integrate accelerometer data [3,4]. The more information passed to an AI system, the easier it is to learn a pattern. However, in practical applications, more sensors mean more computational resources, and usually they may not be available on a device with limited capacities, such as a smartwatch or smartphone [8]. Therefore, developing a solution that utilizes only one type of sensor renders it more appropriate for use in real-world scenarios.

Traditional Machine Learning (ML) necessitates having an expert to extract valuable information from signal sensors. This would require a significant amount of manual

effort [3,4]. Conversely, we have Deep Learning (DL) models that can extract features as good as, if not better than, classical models [3]. A DL model has a large number of hidden layers that enable it to extract high-level features from the data, giving DL models an advantage over conventional machine learning models, which can only learn from hand-crafted features. However, it remains uncertain as to whether the learned features of DL models have the same level of generalizability as hand-crafted features, or if they are dataset-specific.

Convolutional Neural Networks (CNNs) are among the most widely utilized DL models and have emerged as the state-of-the-art method for numerous tasks. In recent years, One-Dimensional Convolutional neural networks (1D CNNs) have been effective in classifying human activity from wearable sensors, such as accelerometers and gyroscopes [3,4,7–9]. One-dimensional CNNs are similar to standard convolutional neural networks, but instead of processing two-dimensional images, they process one-dimensional signals, such as time series data. One-dimensional CNNs can be trained end-to-end to extract features directly from raw data.

AI systems are now making significant decisions on our behalf, including influencing criminal sentencing and curating online content [10,11]. However, without understanding why these systems make certain decisions, people are reluctant to trust and rely on such systems. The term "eXplainable AI" (XAI) is used to describe an artificial intelligence system that can provide a human with an explanation of how it reached its decisions [11]. The main issue with current deep learning models is that they cannot provide explanations for their decisions, making them difficult to trust. Novel techniques, such as Gradient-weighted Class Activation Mapping (grad-CAM), are being developed to improve the interpretability of deep models [8,12].

T-Distributed Stochastic Neighbor Embedding (t-SNE) and Principal Component Analysis (PCA) are machine learning techniques for visualizing high-dimensional data [13–15]. They possess the capability to condense a set of variables into a smaller set of information, while striving to retain as much input information as possible. These methods are often utilized with raw signals, such as images or time series data. However, the potential for combining these techniques with a deep network has yet to be explored.

This paper describes a method for generating visual explanations of features learned by CNNs applied to HAR. By integrating the training model's learned features with the t-SNE technique, we offer several explanations of the model's decision-making process. Moreover, we demonstrate the transferability of the learned features from one dataset to another through the proposed 2D visualization. To the best of our knowledge, this is the first study to present an approach for accurately visualizing learned features in 1D CNNs applied to HAR tasks.

The remainder of this paper is organized as follows. Section 2 introduces the standard protocol used in developing activity recognition applications. Section 3 discusses related works that have utilized the SHO or HAPT databases, the most commonly used XAI methods in HAR, and how t-SNE can be applied. Section 4 describes the methodology, including 1D CNN architectures, the proposed framework, and databases employed. Section 5 presents the metric results achieved for each experiment performed with the SHO and HAPT datasets. Section 6 covers the explainable results. Finally, in Section 7, we conclude our work, presenting some limitations of this study.

2. Activity Recognition Protocol

Most HAR applications utilize the standard Activity Recognition Protocol (ARP) [3,4,16]. However, the activity recognition protocol on HAR applications may differ depending on the specific problem. Usually, ARP consists of six steps: data acquisition, pre-processing, segmentation, feature extraction, and classification and evaluation. We propose a seven-step approach that incorporates explainability. Figure 1 illustrates our proposed protocol.

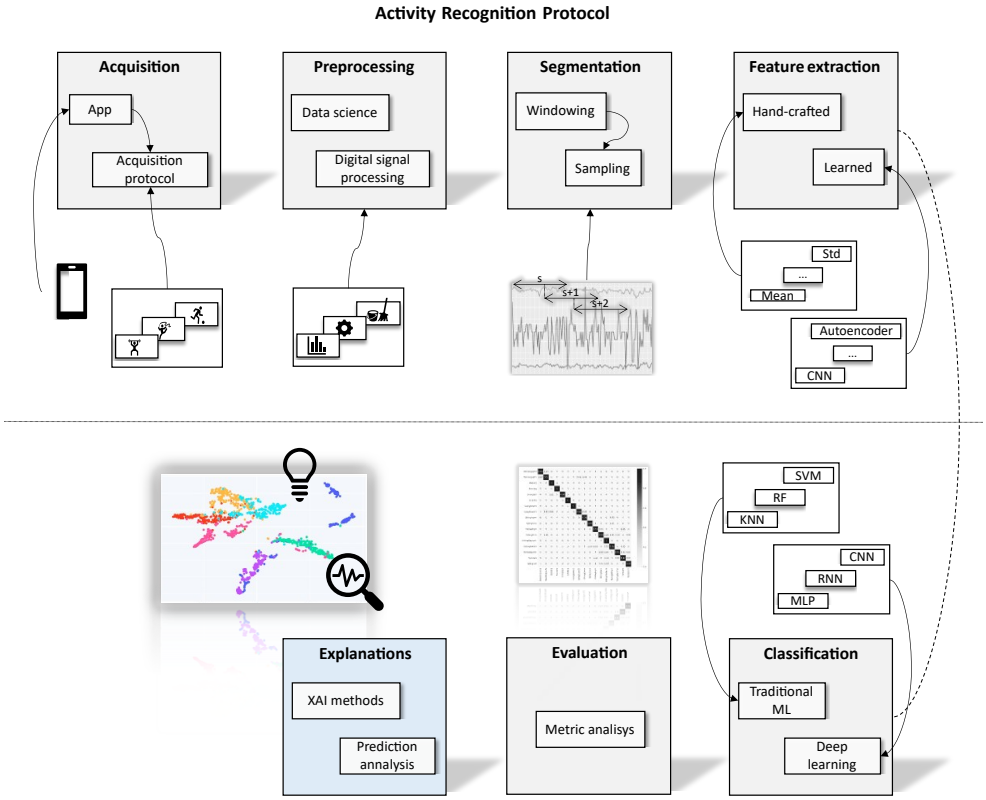


Figure 1. The proposed Activity Recognition Protocol with Explanations includes an additional step for explainability, allowing for analysis beyond metric evaluation. The Classification step utilizes various machine learning algorithms, including Support Vector Machine (SVM), Random Forest (RF), K-nearest neighbors (KNN), Recurrent Neural Network (RNN) and Multilayer Perceptron (MLP).

2.1. Acquisition

In the acquisition step of HAR, data is obtained from sensors via an application that adheres to established acquisition protocols. Some studies also employ a camera or microphone for labeling data [17,18]. Accelerometers are the most frequently utilized sensors in HAR and can measure movement in three directions over time, with a sampling rate of 50 Hz being widely used in literature [3]. Accelerometers can detect diverse activities, including static postures, such as sitting and standing, dynamic activities, such as walking, running, and climbing stairs, and transitional activities, such as standing up, sitting down, and lying down [2–4].

2.2. Preprocessing

Preprocessing in HAR can be performed either during or after acquisition, but it is more common to perform it post-acquisition due to its potential impact on model performance [3]. While sensors may apply basic preprocessing at the hardware level, further preprocessing using digital signal processing, ML, and data science techniques can enhance data quality, identify outliers, and remove noise acquisition [8].

2.3. Segmentation

A robust dataset is crucial for developing a successful AI model, and effective segmentation and labeling of samples are essential [8]. Dataset collection protocols involve continuous data collection, followed by segmentation and labeling of the raw data into smaller segments called windows [8]. Due to variations in activity types and times, effective segmentation can balance the number of samples per class or subject and increase the total number of dataset samples [18,19]. For HAR datasets, the optimal window size is typically three seconds, and event-defined or sliding windowing approaches can be employed to

segment the data [3]. Sliding windowing often employs windows with 50% overlap to ensure consistency between samples [3].

2.4. Feature Extraction

Feature extraction is a crucial step in HAR, involving the transformation of raw data into a reduced set of features that can be more easily processed by machine learning algorithms [2,3]. Two approaches to feature extraction are handcrafted and learned features [2,3]. Handcrafted methods employ mathematical relationships determined by subject matter experts, while learned features are obtained through machine learning algorithms and data correlations. Although deep learning is the most frequently utilized approach for feature extraction in others areas, in HAR many works still employ handcrafted features due to the processing limitations of mobile devices. Handcrafted features can be obtained through statistical attributes in the temporal domain, Fourier transform in the frequency domain, and discretization to obtain symbolic features. In contrast, learned features can be extracted using deep nets, such as CNNs or autoencoders [3]. CNNs extract information through the convolution operation, while autoencoders compress data to extract high-level information in their latent space [8]. The learned features in CNNs are the kernel coefficients, while in autoencoders, they are the latent space of the conversation [8].

2.5. Classification

Classification systems can be approached through either a model-driven or data-driven paradigm [3]. Model-driven approaches attempt to manually reproduce the functioning of the physical system by employing composition rules that can describe the issue as an equation. Data-driven methods employ ML to discover data correlations, and are more commonly used in HAR [3]. ML and DL algorithms utilize statistical and mathematical methods to develop algorithms capable of solving classification problems. Traditional ML classifiers, such as naive Bayes, support vector machines, and decision trees, require feature extraction, since they cannot receive raw data. In contrast, DL systems can handle both simple and complex tasks but require a large amount of data. The algorithm choice can significantly impact performance in classification [8]. Once a classifier architecture is selected, it can be trained in Python using popular frameworks, such as TensorFlow or PyTorch for DL, or Scikit-learn for traditional ML [8,20].

2.6. Evaluation

After selecting a classification algorithm, we must choose a validation technique to split our dataset into training and testing subsets. This stage is necessary to determine if the model has learned to generalize the task by evaluating it with both seen and unseen samples. Two validation strategies are Subject-dependent (SD) and Subject-independent (SI) strategies [2,3,8]. The SI strategy does not use end-user data during training. In contrast, the SD approach takes advantage of this information when training. The performance of the created model during training and validation must be evaluated using a set of metrics. Accuracy, recall, precision, and f1-score are the most commonly employed evaluation measures in HAR, as shown in Table 1 [3,8].

True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the essential elements of all metrics [8] The concept of TP, TN, FP, and FN are based on binary categorization. With a one-vs-all technique, they can be extended to multiclass classification. In this concept, the target class is considered to be positive, while all other classes are combined into a negative class. The most popular metric, accuracy, reflects the proportion of assertiveness overall. Although accuracy is a straightforward metric to comprehend, it does not reflect the true performance of the classifier when we have a dataset with an imbalance of the classes. The precision metric illustrates how the algorithm manages accuracy when predicting positive samples. If we have great precision, our model recognizes TP samples quite effectively. Recall measures how effectively our model deals with false negatives. When there is a strong concern with FN, this metric is applied.

Precision and recall are used to calculate the F1-score. When there is an unbalanced distribution of classes, this measure is crucial.

Table 1. Most commonly used metrics in HAR systems.

Metric	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F1-score	$2 \times \frac{(Precision \times Recall)}{Precision+Recall}$

2.7. Explainability

To better understand a model's decision-making process, we can employ XAI methods. Relying solely on numerical results is not the most effective way to determine the best model [12]. Two models with similar results may have learned completely different features and perform in vastly different ways. Moreover, a model with high performance may still have learned a bias from the dataset [8,12]. Therefore, it is important to use XAI techniques to gain insight into how the model makes decisions and to ensure that the model's predictions are not based on biased or irrelevant features.

There are two distinct approaches on XAI: transparent and post-hoc [11]. Transparent approaches refer to models whose inner architectures and decision-making processes are straightforward and easily understood. Examples of transparent models include the Bayesian model, decision trees, linear regression, and fuzzy inference systems. Transparent approaches are beneficial when internal feature correlations are neither particularly complex nor linear. Post-hoc explainability approaches can reveal the inner workings and decision logic of a trained AI model, providing feature significance scores, rule sets, heat maps, or plain language to assist users in understanding the most relevant information and potential biases [11,21]. In such cases, the post-hoc technique can provide a useful tool for explaining what the model has learned, especially when the relationship between the data and features is not simply a direct one [11].

Within post-hoc methods, we have two distinct approaches: model-agnostic and model-specific [11]. Model-specific approaches provide explainability limitations regarding the learning algorithm and internal structure of a given deep learning model. In contrast, model-agnostic approaches analyze model inputs and predictions in pairs to comprehend the learning mechanisms and generate explanations.

Model-agnostic approaches, such as Local Interpretable Model-agnostic Explanations (LIME) and SHapley additive explanations (SHAP) can be used with handcrafted features and classical ML to determine the importance of input features [22–24].

Model-specific approaches are used in Explainable AI (XAI) to explain the decisions made by a specific machine learning model. These approaches are tailored to the model's architecture, parameters, and training process, and they often rely on knowledge of the internal workings of the model. Examples of model-specific approaches include decision tree induction, rule extraction, gradient-based methods, and layer-wise relevance propagation [12,25–27].

These approaches provide valuable insights into the model's decision-making process and can help improve the model's performance and reduce biases.

3. Related Works

This section discusses related studies in the HAR area, focusing on those that use SHO or HAPT datasets. Next, we discuss the studies that used some XAI techniques in the HAR area. Finally, we discuss the t-SNE technique and applications.

3.1. Related Studies in HAR

The earliest implementation of HAR dates back to the 1990s. Since then, numerous strategies and frameworks have been proposed. HAR may utilize different sensors, including cameras, piezoelectric, GPS, microphones, and inertial measurement units. Moreover, several publicly available datasets, including HAPT, UniMiB-SHAR, SHO, and WISDM [3,17,18,28], have been developed to aid research in this field.

CNN architectures are frequently employed in HAR due to their scalability and transition invariance, which contribute to their robustness. Most CNN designs are encoder-like, causing the data to lose dimensionality as it progresses through the layers, leading to the acquisition of advanced semantic meanings. These structures can extract representative features, as well as incorporate handcrafted approaches [29]. Multidimensional data can serve as inputs, as in the case of a nine-dimensional signal composed of a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer. CNNs are not limited to raw data. They can also receive manually created features or features learned from other deep networks. HAR-Net, for instance, is a CNN architecture developed by Dong and Han [30], capable of receiving customized input characteristics. Juefei-Xu et al. [31] calculated the vector sum of accelerometer signals in the x, y, and z axes to create a magnitude signal, which they inputted into a CNN, resulting in superior performance compared to a random forest approach. Ronao and Cho [32] proposed a CNN architecture that employed six-dimensional data as input, consisting of a tri-axial accelerometer and a tri-axial gyroscope, achieving an accuracy of 0.94.

Table 2 demonstrates the frequent utilizations of the SHO and HAPT databases in recent papers. Research on the usage of the HAPT database reveals that some authors choose to group the Postural Transition (PT) classes from the dataset. This grouping strategy may involve dividing the PTs into a single subgroup [18,33] or two subgroups [33,34]. However, there are only a limited number of studies that attempt to classify the PTs into 12 distinct classes.

Regarding works that have utilized the HAPT database, Reyes-Ortiz et al. [18] introduced it by combining an existing database primarily consisting of positional transitions. The authors developed a system that employs a support vector machine classifier and a set of heuristic rules to classify six types of human activities. In their article, they grouped all postural transition activities into a single class, demonstrating that this strategy improved recognition performance of the six Activities of Daily Living (ADLs) considered. Thu and Han [34] created the HiHAR framework, which employs two deep architectures, Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory Networks (BiLSTM), to classify PT activities into two subgroups [33]. Thu and Han [33] evaluated their architectures considering PTs as a subgroup, two subgroups, and as separate activities.

In their work on the SHO dataset, Jiang and Yin [35] proposed a framework that utilized input from accelerometer, gyroscope, and linear acceleration sensors. The temporal data was converted into an image through a process that utilized the Discrete Fourier Transform (DFT). The framework employed the eight activities available in SHO, but only utilized data from a sensor placed on the wrist. On the other hand, [16] considered all available sensor positions, but only six activities. The activities were selected to be compatible with the physical activities used in other datasets in their study.

Table 2. Performance comparison of different HAR models using the HAPT and SHO datasets; N × M—N is the input size and M is the input dimension; FE—feature extracted; Acc—Accelerometer; Gyro—gyroscope; BA—basic activities; PT—postural transition; SD—subject-dependent; SI—subject-independent; CV—cross-validation; LOSO—Leave-one-subject-out.

Author	Dataset	Model	Input (N × M)	Validation Strategy	Classes	Best Performance
Reyes-Ortiz et al. [18]	TAHAR system with SVM	561 × 1 FE vector (Acc + Gyro)	N/A	No validation subset	6 BA + 1 PT group	0.9700 accuracy
Thu and Han [34]	HAPT	HiHAR-8 and CNN	128 × 6 (Acc + Gyro)	SI: 24/6	6 BA + 2 PT groups	HiHAR-8: 0.9798 accuracy CNN: 0.9440 accuracy
Thu and Han [33]	HAPT	BiLSTM * and CNN	No info about input shape. FE: DWT ** (Acc + Gyro)	SD: Hold-out 80/20	6 BA + 1 PT group	BiLSTM: 0.9634 accuracy
					6 BA + 6 PT	BiLSTM: 0.9487 accuracy
						CNN: 0.9171 accuracy
Jiang and Yin [35]	SHO, only wrist sensor position	2D CNN	36 × 68 - image generated from DFT *** (Acc + Gyro + Linear acceleration)	SD: Hold-out 70/30	8 BA	0.9993 accuracy
Braganca et. al. [16]	SHO - all available sensor positions	Random forest	64 × 1 FE (Acc)	SD: 10-CV, SI: LOSO	6 BA	SD: 0.98 MAA ****, SI: 0.8412 MAA

* BiLSTM = Bidirectional Long Short-Term Memory; ** DWT = Discrete Wavelet Transform; *** DFT = Discrete Fourier Transform; **** MAA = Mean Average Accuracy

3.2. Explainable AI with Timeseries

In recent years, AI has been widely adopted across industries and applications. As a result, there is a growing demand for XAI systems that can offer transparent explanations for their decisions and actions. This is especially true in the analysis of time-series data, where interpreting complex patterns and trends over time can be challenging [36,37]. In addition, time-series data is frequently obtained from multiple sensors, generating massive amounts of information that can be difficult to interpret without the appropriate context [37].

In time-series data, XAI techniques can be particularly valuable for understanding complex patterns and trends that evolve over time. Gradient, structure, and surrogate techniques offer different approaches to generating attributions, which can help in explaining AI model decisions [21,38]. Gradient methods are often an efficient starting point for generating attributions, given their computational speed for single samples. On the other hand, structure methods use the learned network weights and biases to propagate a score from the output to the input. Finally, surrogate and sampling methods, such as Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP), create perturbed instances of the input sample and train an interpretable or game-theoretical model to generate attribution scores.

Assaf and Schumann [39] present a dense-pixel visualization technique for time series data by displaying each time point as a rectangle with the relevance score represented by color. This approach allows for investigating patterns and explaining complex model

decisions. However, using grad-CAM [12] for attributions leads to some patterns which are difficult to understand, even with domain expertise. [36] modify this approach by changing the color scale and replacing grad-CAM [12] with a CAM-like algorithm that focuses more on neural network architecture, resulting in improved attributions. Despite these improvements, the visualization still needs to provide insights into the model. Furthermore, it can be challenging to interpret, due to the selected color scale.

Schlegel and Keim [38] combine the approaches of Assaf and Schumann [39] and Viton et al. [36] by overlaying a line plot of the time series data on a background heatmap representing attributions. This approach aims to include both time series and attribution data. Still, it can be more difficult to understand, especially for non-experts.

3.3. Explainable AI in HAR

Explainable AI (XAI) refers to a model's capacity to provide justifications or explanations for its predictions. The majority of XAI techniques [40,41] have been presented for video-based activity recognition. However, generating meaningful explanations for predictions derived from sensor data is more complicated. Therefore, XAI in the context of being HAR sensor-based would involve understanding how the model identifies diverse activities from sensor data and what specific features or patterns it employs to make predictions. This can help enhance the model's interpretability and transparency and identify and correct any potential biases or errors in the model's predictions.

Existing research efforts on explainable techniques for sensor-based activity identification are limited, and they examine only intrinsically interpretable models [42]. Bettini et al. [43] claim that, despite the high performance of existing methods for classifying human activities, it is difficult to find a rational explanation of what the model considers to make predictions. Referring to the term XAI, they proposed a framework named XAR, with the letters AR referring to activity recognition. The entire framework was focused on interpretable models, and, in this case, random forest and support vector machines were utilized. They generated explanations by combining the feature values with the feature importance derived from the classifier's training. Their analysis of a real-world ADL dataset demonstrated that the method successfully provides explanations consistent with conventional wisdom. Furthermore, the proposed framework enables non-expert users with its intuitive and easily understandable explanations of detected movement or activity through natural language.

The authors in [44] utilized a classifier based on rules. In the training step, the model acquires a set of rules that encapsulate the correlations between sensor events and actions and are legible by humans. The findings suggested that the proposed model achieves recognition rates comparable to those of established interpretable classifiers (e.g., Decision Tree, JRip), while producing considerably less complicated rules. In their study, Arrotta et al. [42] introduced DeXAR, an advanced XAI framework that aims to improve the detection of ADLs using DL algorithms. The framework utilizes sensor data and transforms it into semantic images, enabling 2D XAI methods for ADL detection. Furthermore, the authors implemented various XAI algorithms for deep learning, providing non-expert users with natural language explanations based on the resulting heat maps. Finally, to evaluate the effectiveness of the proposed XAI methods, they conducted a thorough evaluation of two different datasets, using both traditional common-knowledge evaluation and user-based evaluation methods.

The authors in [8] proposed employing gradient-weighted class activation mapping (grad-CAM) to generate visual explanations for the decision-making process of deep learning models utilized for HAR and biometric user identification tasks, using accelerometer data. The study found that the high performance of HAR using SD validation was not solely based on physical activity learning but also on learning an individual-specific signature. Overall, the paper suggests that combining explainable techniques with deep learning can facilitate the design of better models, while mitigating overestimation of results. These models can provide a more coherent interpretation of the model's decision-making processes and can be easily visualized to help understand how the model operates.

This paper presents a novel approach to activity recognition by leveraging the capabilities of XAI techniques. Specifically, we propose a t-SNE-based visualization technique to access the quality of features learned by deep learning models and to identify confusion and misclassification between samples. This approach offers an in-depth understanding of the generalization capabilities of DL models. It has the potential to assist AI researchers in developing more effective models and in understanding their limitations in real-world scenarios.

3.4. T-Distributed Stochastic Neighborhood Embedding

T-distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction technique that performs well for visualization, allowing a projection of the data in low-dimensional spaces that make it simple to apply to very large datasets [13,14]. Using t-SNE during the learning process reduces the dimensionality of the dataset, while preserving its topology, by improving clustering accuracy.

The t-SNE technique utilizes Stochastic Neighbor Embedding (SNE) [14]. The method utilized by both approaches converts high-dimensional data into a probability distribution. Then, it maps this distribution onto a low-dimensional space, such as a 2D plane or a 3D space. Principally, the SNE converts high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities. The t-SNE technique utilizes the student t-distribution with one degree of freedom as the heavy-tailed distribution in the low-dimensional map.

The t-SNE algorithm begins by calculating the pairwise similarities between all high-dimensional data points [14]. These similarities are represented as a high-dimensional probability distribution, a probability distribution with a high number of dimensions. Then, t-SNE generates a low-dimensional probability distribution by assigning a probability to each pair of points in the low-dimensional space. This low-dimensional probability distribution is intended to closely resemble its high-dimensional counterpart.

Kullback–Leibler divergence is the t-SNE cost function, which measures the difference between the high-dimensional and low-dimensional probability distributions [14]. The cost function is defined as:

$$C_{t-\mathrm{SNE}} = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (1)$$

Kullback–Leibler divergence (KL) is a measure of the difference between two probability distributions. In the t-SNE algorithm, KL divergence is used as the cost function, where P and Q represent the high-dimensional and low-dimensional probability distributions, respectively. In Equation (1), p_{ij} denotes the probability that the high-dimensional point i is selected as the neighbor of point j, whereas q_{ij} is the probability that the low-dimensional point i is selected as the neighbor of point j.

Through adjusting the positions of the points in the low-dimensional space, t-SNE seeks to minimize this cost function. The gradient of the cost function with respect to the low-dimensional coordinates is typically calculated and used to update the coordinates using a gradient descent algorithm.

In HAR, t-SNE is usually used to visualize the separation of the data. Dharavath et al. [45] used t-SNE to visualize the data distribution in a HAR dataset. The dataset contained an accelerometer and gyroscope readings while performing activities such as walking, walking upstairs, walking downstairs, standing, and lying. The author ran the raw data through the t-SNE method, comparing the generated visualization with a visualization obtained with the PCA technique. The visualization with t-SNE was significantly better.

Thakur et al. [46] used PCA to extract the characteristics of human activities from both accelerometer and gyroscope sensors. The obtained features were input into t-SNE to visualize the data's separation. The extracted features from PCA were also fed into an ensemble of three ML classifiers to identify six distinct human physical activities.

4. Methodology

This section explains the architectures implemented, the public database used and the proposed framework. In practical applications, more sensors mean more computational resources, which are usually not available on a device with limited capacities, such as a smartwatch or smartphone [8]. Therefore, developing a solution that utilizes only one type of sensor renders it more appropriate for use in real-world scenarios.

4.1. Architectures

The simulations were conducted utilizing CNN1 and CNN2 convolutional network architectures, initially introduced by Aquino et al. [8]. The models were trained and implemented using the TensorFlow 2 framework [20]. Initially, we emphasized the commonalities between both designs. In both instances, the Adam optimizer was employed with its default hyperparameters. The models underwent 300 training epochs. Each convolutional block comprised two layers, each of which contained 100 feature maps. The ReLU activation function was utilized. In every max-pooling layer, the pool size and stride were set to 2. For dropout, a value of 0.5 was applied [47]. The Softmax layer contained n neurons, where n represents the number of classes for the problem. A custom callback was employed to create checkpoints after each training epoch, selecting the optimal model based on the macro F1-score metric for the validation set. The batch size was set to 256. Both architectures received identical 151 $\times$ 3 signals at their inputs.

The differences between the two architectures are now discussed. The first architecture comprised four convolutional blocks, each with two convolutional layers, while the second architecture incorporated only three convolutional blocks. Furthermore, CNN1's kernel size was eight, whereas CNN2's was four. Figure 2 illustrates both architectures.

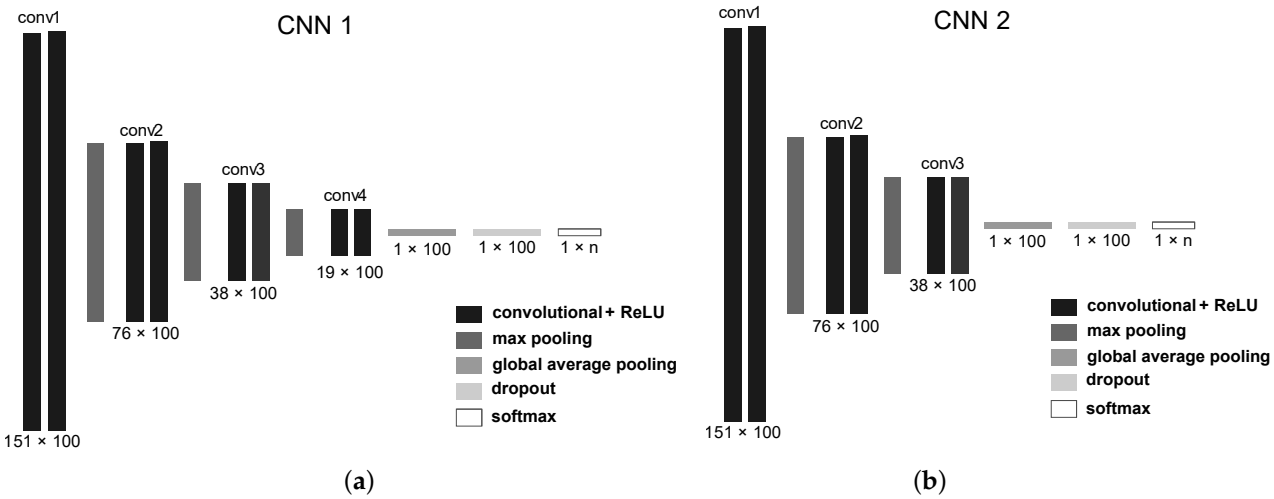


Figure 2. The CNN architectures implemented with TensorFlow 2. (**a**) shows the CNN1 architecture with 4 convolutional blocks. (**b**) shows the CNN2 architecture with 3 convolutional blocks.

4.2. Databases

This section provides specific information about the datasets used in this study. Human activity recognition (HAR) can employ various sensors, such as accelerometers, gyroscopes, and magnetometers. Accelerometers are highly effective in detecting movements, and many related studies have used this type of sensor alone, due to its relevance and reliability in HAR [3,4]. Although combining sensors can improve accuracy, this may not be feasible for devices with limited processing capacity, such as smartphones or smartwatches [3,8]. Therefore, using a single sensor, such as an accelerometer, is a more practical approach for real-world HAR applications [8,9,48].

4.2.1. SHO

The authors collected data for seven physical activities: walking, running, sitting, standing, jogging, biking, walking upstairs, and walking downstairs [28]. Ten volunteers performed each task for 3–4 min as part of the data capture project. All ten subjects were men between the ages of 25 and 30. Except for cycling, the studies were conducted inside a university facility. Each of these participants was equipped with five smartphones for use in five different body postures: right pocket, left pocket, belt position toward the right leg using a belt clip, right upper arm, right wrist.

For the trials, Samsung Galaxy SII (i9100) smartphones were used. For each movement, data was collected at a rate of 50 samples per second for all five places simultaneously. Accelerometer, gyroscope, magnetometer, and linear acceleration sensor information were collected [28].

In this work, for the dataset generation, a 3-s window was used, with 50% overlapping. The data were collected only from the waist position using a belt, and only the accelerometer sensor was considered. For the subject-dependent (SD) method, we implemented a randomized partitioning strategy with shuffling, adhering to a conventional 70/30 split for training and validation. Additionally, for the subject-independent (SI) approach, we endeavored to maintain the same proportion in the partitioning, yielding 7 subjects for training and 3 subjects for validation. Validation was performed using subjects 1, 2, and 3.

SHO is a high balanced dataset. For our setup, a total of 4130 samples was obtained, with exactly 590 samples per class, and all ten subjects provided the same number of samples.

4.2.2. HAPT

The UCI Human Activity Recognition Using Smartphones Dataset was expanded in this dataset [18]. Instead of the pre-processed signals from the smartphone sensors that were supplied in version 1, this version offered the original unprocessed raw inertial signals from the sensors. Additionally, the activity labels were revised to incorporate postural changes that were absent from the earlier dataset version.

The public HAPT dataset was obtained from thirty participants ranging in age from 19 to 48 years [18]. The dataset contained raw inertial signals obtained from 3-axial linear acceleration and 3-axial angular velocity sensors integrated in a smartphone equipped at the waist by the user. It included 6 basic activities (BAs): standing, sitting, laying, walking, walking upstairs, and walking downstairs#, and 6 postural transitions (PTs) between three static postures, stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand.

The data was collected at a constant sampling frequency of 50 Hz. Signals were then synced with experiment recordings so that they could be used as the ground truth for hand labeling [18].

For dataset generation the same setup used in SHO dataset was applied, namely, a 3-s time window with 50% overlapping, and only an accelerometer sensor. The data were collected only from the waist position using a belt. Table 3 displays more details about the data distribution of the dataset. As shown in the N° Sample column, HAPT was an unbalanced dataset. Finally, as shown in the N° subjects' column, we can see that not all participants performed all activities, especially for PT.

For the subject-dependent (SD) method, we implemented a randomized partitioning strategy with shuffling, adhering to a conventional 70/30 split for training and validation. In the SI approach, we took into account that not all individuals performed all physical activities. Therefore, for the validation subset, the first nine individuals, who performed all activities, were chosen: subjects 1, 2, 3, 4, 5, 6, 13, 17, and 18. The remaining subjects were used in the training subset.

Table 3. Activity and postural transition (PT) classes in the dataset with corresponding labels, number of samples, and number of subjects.

Action	Label	N° Samples	N° Subjects
BA	Standing	855	30
	Sitting	782	30
	Laying	851	30
	Walking	746	30
	Walking Upstairs	687	30
	Walking Downstairs	614	30
PT	Stand-to-sit	39	24
	Sit-to-stand	14	10
	Sit-to-lie	59	30
	Lie-to-sit	51	28
	Stand-to-lie	71	29
	Lie-to-stand	50	29

4.3. Proposed Framework

To summarize, we propose the framework illustrated in Figure 3. In the dataset building step, we loaded the SHO and HAPT databases and performed an exploratory analysis of the data.

This exploratory analysis allowed us, for example, to identify that not all individuals performed all activities in the HAPT database. We also defined a window size of 3 s with 50% overlapping. After this, the data was subdivided in the splitting step according to SD or SI validation strategy. Then, the models were implemented and evaluated, with in training and evaluation steps, and their results contrasted using performance metrics. Again, we used the CNN1 and CNN2 architectures. Finally, we conducted an in viewing learned features step, in which we obtained the embeddings for the architectures, and used this as input for the t-SNE. By doing this, we had a visualization that clearly explained how the model performed the data distribution from the learned features.

In this work, we proposed using a framework to visualize the power of features learned from a dataset or how features learned from one dataset behave in another unseen dataset. For evaluation on the same trained dataset we, first, used the dataset to train the two CNN architectures, CNN1 and CNN2. We chose the best architecture based on the achieved numerical results. We obtained XAI visualizations and provided explanations of the decision-making process.

In other words, the HAPT samples were passed through this network, and a vector with information from one layer before classification was obtained. After that, new visualizations were performed, which allowed us to conclude the limitations of the learned features.

The proposed framework allows for the visualization of the effectiveness of features learned within a single dataset and the examination of their performance on an unseen dataset. This is achieved by implementing the proposed activity recognition protocol and including a visual explanation step to gain insight into the ability of learned features to differentiate activities on another dataset.

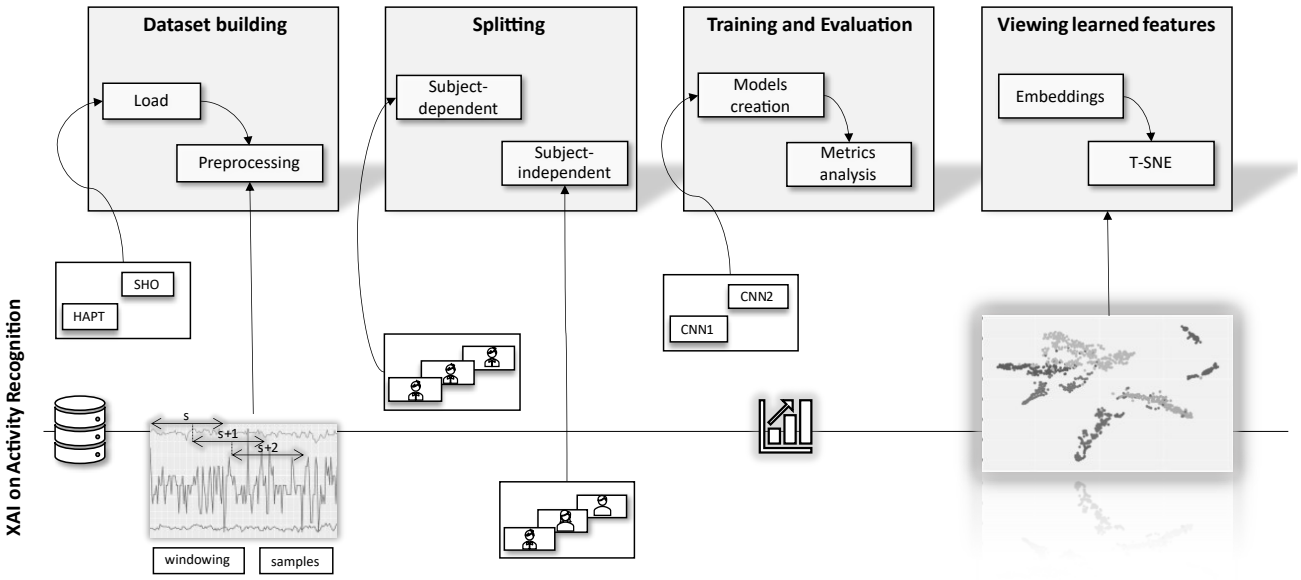


Figure 3. A schematic representation of the proposed framework for XAI on activity recognition. The framework consists of four steps: dataset building, splitting, training and evaluation, and viewing learned features with our proposed t-SNE visualization technique, where each color represents a different activity.

5. Metric Results

5.1. SHO

In this study, we evaluated and trained two CNN architectures, CNN1 and CNN2, employing both the SD and SI strategies. To enable a more precise comparison with other relevant studies, we utilized both macro-average and weighted-average metrics. The macro-average metric considers the impact of the metric on each class individually, while the weighted-average metric accounts for the metric's results on each class, weighted by the number of samples assessed. The macro-average metric is especially beneficial for handling imbalanced systems, as it does not discriminate between classes with larger data [8]. Conversely, the weighted-average metric offers a more comprehensive understanding of the system, assuming that the class distribution in the dataset resembles that of real-world data [8]. Detailed performance results for each network, considering both the SD and SI strategies, are presented in Tables 4 and 5.

Table 4. Classification performance of CNN1 and CNN2 in HAR, with the SD strategy and 70/30 split on SHO dataset.

Model	Macro Average			Weighted Average			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
CNN2	0.9976	0.9976	0.9976	0.9976	0.9976	0.9976	0.9976
CNN1	0.9966	0.9969	0.9967	0.9968	0.9968	0.9968	0.9968

Table 5. Classification performance of CNN1 and CNN2 in HAR, with the SI strategy and 3 subjects on validation.

Model	Macro Average			Weighted Average			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
CNN2	0.9816	0.9814	0.9814	0.9816	0.9814	0.9813	0.9814
CNN1	0.9786	0.9782	0.9781	0.9786	0.9782	0.9781	0.9782

The results obtained from the SD validation strategy were marginally superior to those acquired with the SI strategy for both trained architectures, across all metrics. This outcome was expected based on prior research, which suggests that the SI validation method is more rigorous and offers a more accurate depiction of the network's performance in real-world scenarios, as the network is not provided with prior knowledge about how individuals perform an activity [4].

The CNN2 architecture outperformed CNN1 using the SD strategy, while, with the SI strategy, CNN1 performed better. The performances of both architectures were closely matched, with a difference of only 0.02% in the SD strategy and 0.3% in the SI strategy.

The SI approach was determined to be more demanding and better aligned with how the model would be evaluated in real-world scenarios. Consequently, greater emphasis was placed on the SI results. By examining the confusion matrix depicted in Figure 4, we gained insight into the challenges encountered by the CNN1 model when utilizing the SI approach.

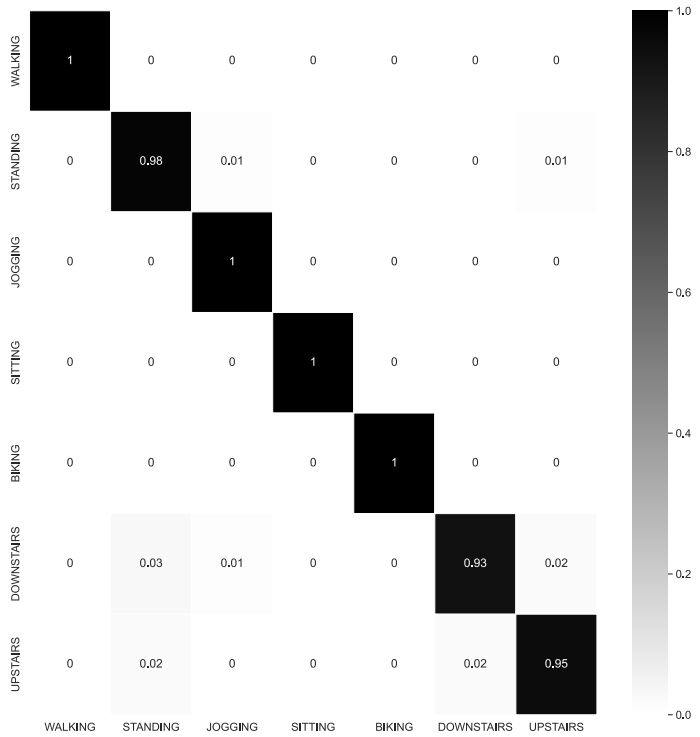


Figure 4. Confusion Matrix of CNN1 with subject-independent, with 3 subjects in validation for all 8 classes in SHO dataset.

5.2. HAPT

Since this database was highly imbalanced, the results were anticipated to be less remarkable, given that there were 12 classes and only a few samples of the postural transition activities. The results are displayed in Tables 6 and 7.

Table 6. Classification performance of CNN1 and CNN2 in HAR, with the SD strategy and 70/30 split.

Model	Macro Average			Weighted Average			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
CNN2	0.8724	0.8817	0.8742	0.9515	0.9509	0.9508	0.9509
CNN1	0.8749	0.8706	0.8723	0.9482	0.9481	0.9406	0.9481

Table 7. Classification performance of CNN1 and CNN2 in HAR, with the SI strategy and 9 subjects on validation.

Model	Macro Average			Weighted Average			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
CNN2	0.8734	0.8632	0.8638	0.9289	0.9275	0.9274	0.9275
CNN1	0.8624	0.8603	0.8546	0.9259	0.9239	0.9238	0.9239

The results obtained between the two approaches were comparable. Considering the macro F1-score, there was a difference of less than 1% between the best architecture using the SI and SD approaches. This difference was under 3% in terms of the accuracy metric. The weighted metrics achieved the best results in both approaches, as anticipated, due to the imbalanced nature of the dataset.

Figure 5 displays a confusion matrix for the best architecture utilizing the SI approach.

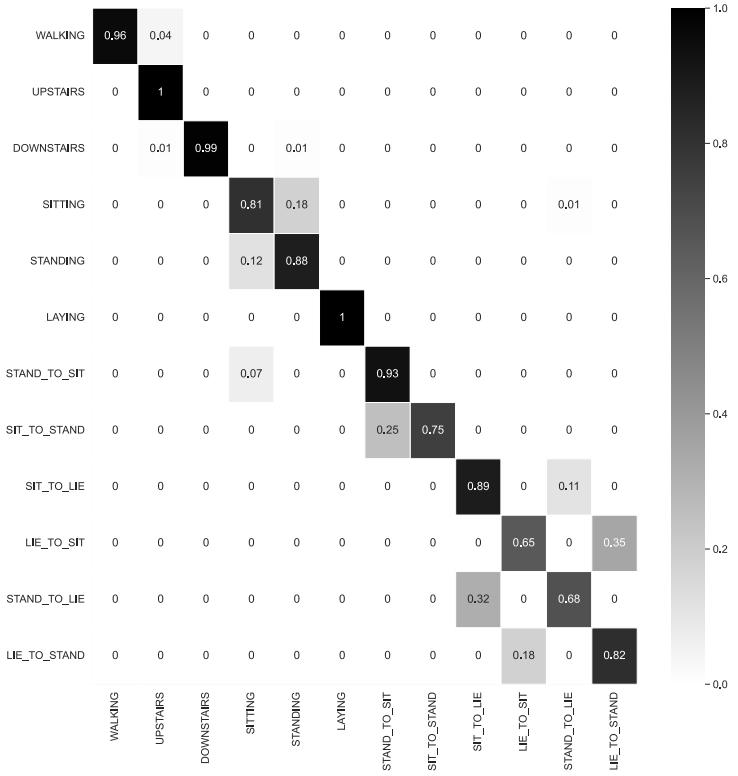


Figure 5. Confusion Matrix of CNN1 for SI approach, with 9 subjects in validation set, using all 12 classes in HAPT dataset.

6. XAI Results

In the XAI results section, we explore the outcomes obtained with the deep network CNN1 on the SHO and HAPT datasets, employing the proposed t-SNE visualization through the learned features. The results are scrutinized and interpreted to emphasize the efficacy of t-SNE in detecting bias issues and identifying mislabeled samples. The visualization further illustrates the deep network's generalization capability in distinguishing activities and its constraints when applied to a distinct dataset. Ultimately, the discussion offers insights into how t-SNE visualization can augment the comprehension of deep networks in HAR tasks.

6.1. SHO

Following the training of the models and evaluation of the numerical results, we proposed employing XAI to generate visual representations that elucidate the model's learning process. We extract the model's embeddings up to one layer before the Softmax, and, subsequently, trained a t-SNE based on this information. The outcome was a two-dimensional embedding that endeavored to preserve the information present in the model's output. This approach enabled clear visualization of the model's acquired knowledge regarding data characteristics and how the data was partitioned, based on the learned features. Figure 6 presents the results graph.

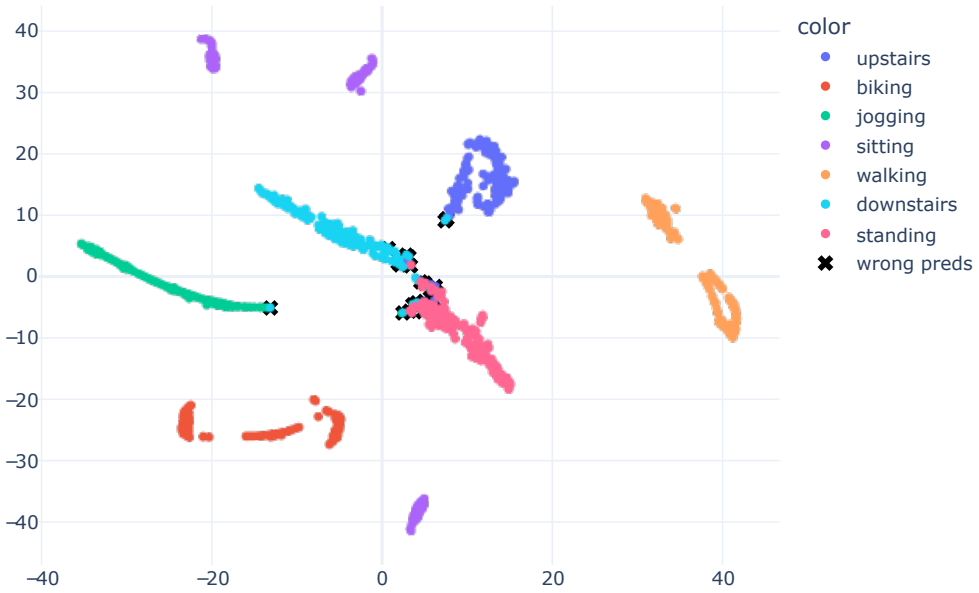


Figure 6. T-SNE Embedding Plot obtained from the CNN1 architecture with HAR using the SI approach for the SHO dataset. We utilized 40 perplexity and conducted 500 iterations.

The visual results obtained corresponded with the numerical results presented in Table 5, wherein the model achieved exceptional performance with an accuracy of 0.98. In the plot, the data from different classes were dispersed. Confusion occurred between activities, such as downstairs and standing. This outcome was also evident in the confusion matrix shown in Figure 4, where the downstairs class attained lower precision than other activities.

As depicted in Figure 6, black x marks were drawn on samples with incorrect predictions to identify the model's errors. The resulting representation enabled verification that the incorrect predictions were readily observed and explained. For instance, we examined the misclassified sample within the jogging cluster, indicated by a black x. The ground truth for this sample was the downstairs class. This confusion arose due to the features learned by the model, which positioned this sample near the jogging cluster. This may have occurred as a result of a limitation in the trained model, which might have learned characteristics that did not optimally separate the data. Alternatively, this sample could have been mislabeled. Nonetheless, the obtained visualization allowed for an understanding of why the model made an error.

In the visualization, samples from downstairs were more dispersed, indicating that the downstairs class was the most challenging in the dataset. The model may be more susceptible to predicting this class in real-world scenarios. Moreover, by analyzing the results of the obtained graph, it became evident that there were subgroups within the same class, such as sitting, which featured three distinct subgroups scattered throughout the view. Similarly, even walking had two subgroups. These analyses were not possible solely

through numerical results, and the visualization presented various opportunities to explain the model's predictions.

An intriguing observation was made by analyzing the sitting activity. There were three clusters, which corresponded to the three different subjects present in the validation subset. This finding suggested that the learned features may have the potential to not only distinguish the sitting activity, but also differentiate the subjects.

To further highlight the capabilities of this visualization, Figure 7 displays a visualization using PCA. The method applied was similar to t-SNE. Initially, we utilized the model's embeddings output, one layer before the classification, as input for the PCA. Next, we applied PCA to derive only three components, which were employed to generate the plot. We calculated merely three components to demonstrate that there was no combination of components for which the resulting visualization was as informative as the one obtained with t-SNE.

It was anticipated that the data would exhibit clear class separation; however, this visualization did not align with the numerical results. Providing explanations was challenging, since, in this visualization, the majority of classes were situated close to each other. This outcome might have arisen because PCA lacks the capability to extract relevant information from non-linear data, which could have been the case with the features learned by the model.

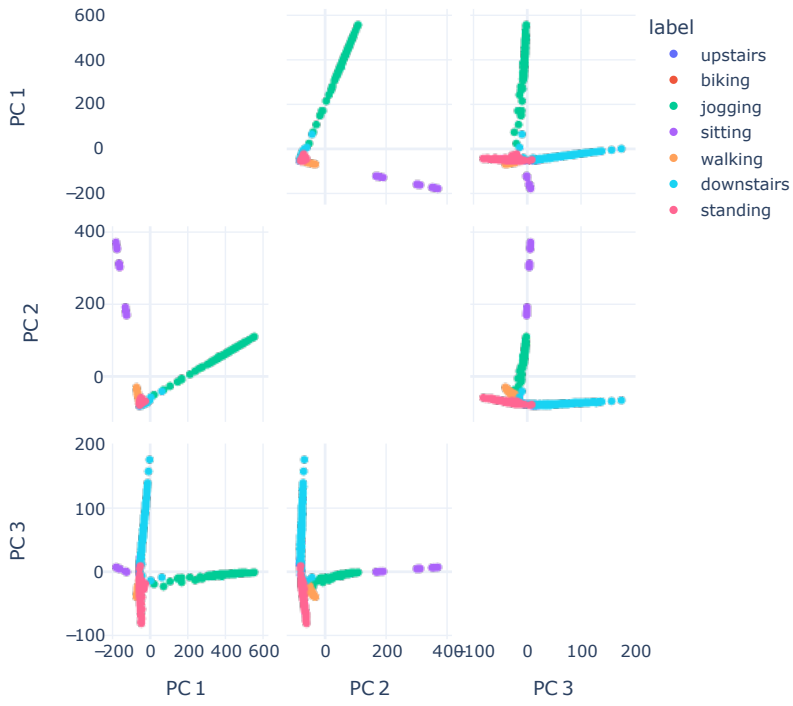


Figure 7. Embeddings visualization with PCA, combining three components, 2 by 2.

The representation in Figure 6 demonstrates how the model learned to separate the data based on the features it acquired. We plotted the graph displayed in Figure 8 to analyze the distribution of raw accelerometer data. To generate this plot, we concatenated the data from the X, Y, and Z axes of the accelerometer and used it as input for the t-SNE algorithm. Our work introduces the innovation of applying t-SNE one layer before the model's classification, setting it apart from the standard approach of applying t-SNE on raw data, as seen in previous works [13,14,45,46].

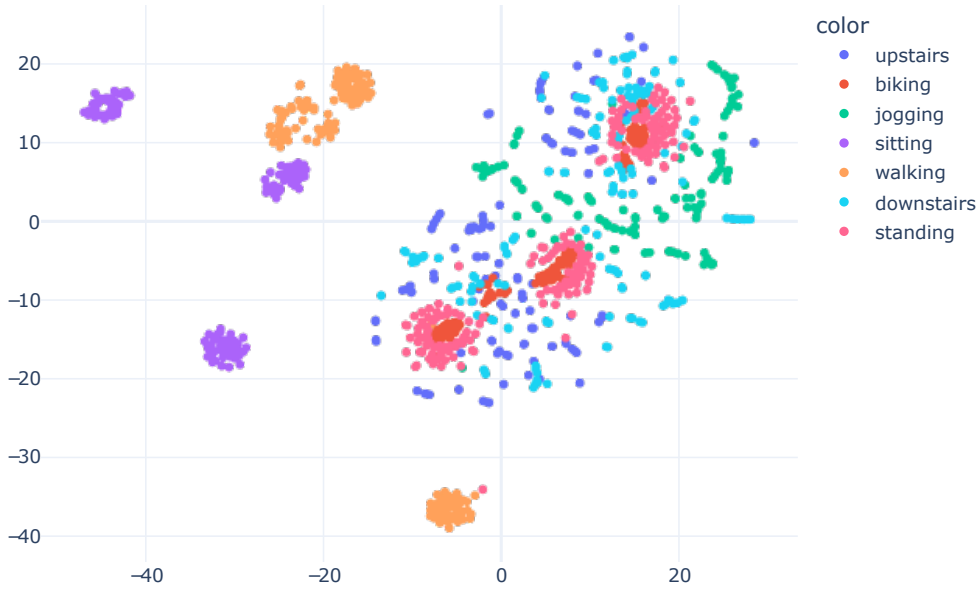


Figure 8. SHO accelerometer raw data visualization with t-SNE, concatenating X, Y and Z information. We utilized 40 perplexity and conducted 500 iterations.

By visualizing the plot of the learned features during the model's training, it became evident that the model underwent a transformation in regard to the input data. In its original form, the raw data presented a challenge in differentiating between human activities. However, after the model learned and distilled the relevant characteristics, the separation between classes became more evident and distinguishable. This visualization offered valuable insights into the model's inner workings and assisted in enhancing its performance.

6.2. HAPT

For the HAPT dataset, based on the results presented in Table 7, a more significant confusion between samples was expected compared to the representation obtained with the SHO dataset. Examining Figure 9, it is possible to observe a lesser separation between samples of different classes. The main issue lay in the Postural Transitions (PT) classes. This result was also anticipated when observing the confusion matrix in Figure 5.

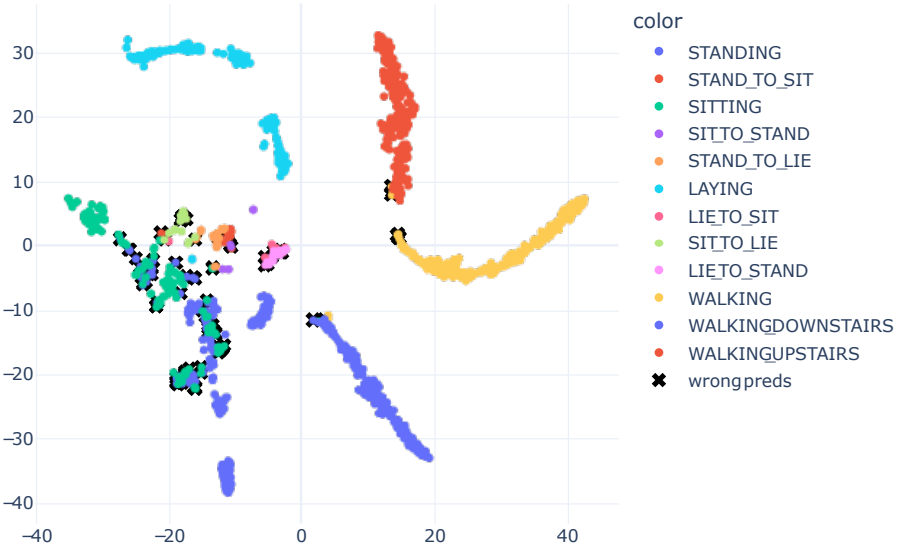


Figure 9. HAPT t-SNE visualization. We utilized 40 perplexity and conducted 500 iterations.

To demonstrate that the model learned relevant features for the other activities, excluding Postural Transitions (PTs), Figure 10 is presented. Other works have already considered these PTs as one or two subgroups. Some studies only use PTs to enhance the performance in recognizing other activities.

The model encountered a familiar challenge, similar to the SHO dataset, with the standing class proving difficult to classify. However, in contrast to the SHO dataset, confusion arose between standing and sitting classes. Further analysis revealed the presence of subgroups within the laying, standing, and sitting classes.

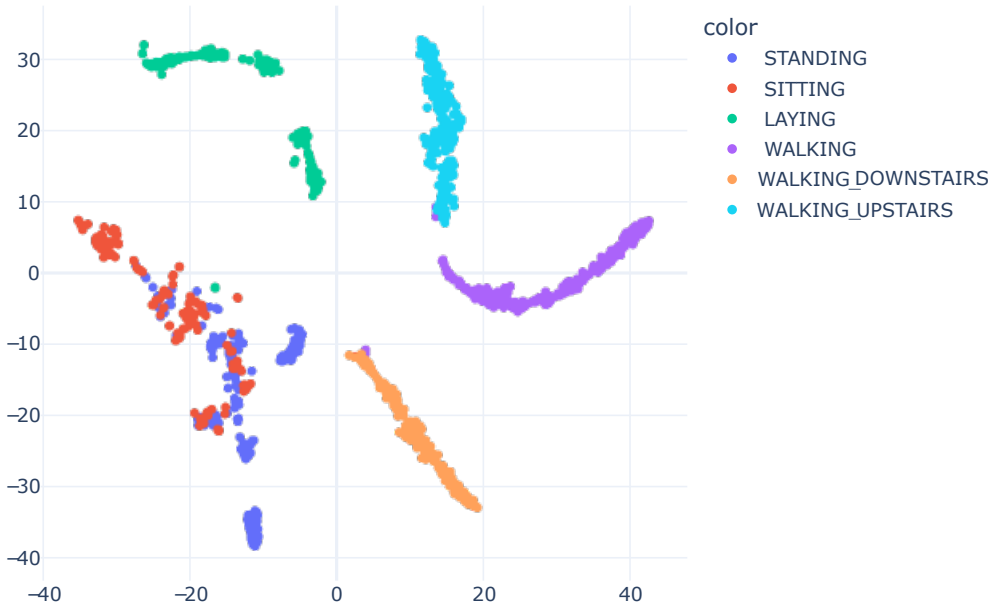


Figure 10. T-SNE Embedding Plot obtained from the CNN1 architecture with HAR using the SI approach for HAPT dataset, excluding PT activities. We utilized 40 perplexity and conducted 500 iterations.

To better illustrate the original data distribution, Figure 11 displays the raw accelerometer data using t-SNE. The same process utilized to generate Figure 3 was applied to the HAPT dataset.

Figure 11. Accelerometer raw data visualization with t-SNE, concatenating X, Y and Z information. We utilized 40 perplexity and conducted 500 iterations.

As occurred with SHO, utilizing raw data as t-SNE input was a challenge because there were no highly relevant features, in contrast to Figure 9, where the features were those learned by the neural network and had mathematical relevance.

Overall, the models demonstrated their ability to learn meaningful features, as evidenced by the t-SNE visualizations and performance metrics obtained on both the SHO and HAPT test sets. Next, we assess whether the features learned in one dataset could accurately classify human activities in another dataset.

6.3. SHO Features into HAPT

To analyze the relevance of the features, we used a network trained on the SHO dataset to extract features from the HAPT dataset. To do this, we performed a prediction with the SHO model, propagating the HAPT samples and considering the result before the classification layer. This result was used as input to the t-SNE algorithm.

Figure 12 shows the resulting representation. However, the SHO dataset's learned features needed to be universal to be useful for other datasets. In this case, the SHO features caused the pattern of the HAPT dataset to be more confusing than the raw data. Although Figure 6 successfully distinguished between activity types using the SHO dataset as a reference, this approach produced unsatisfactory results when applied to the HAPT dataset.

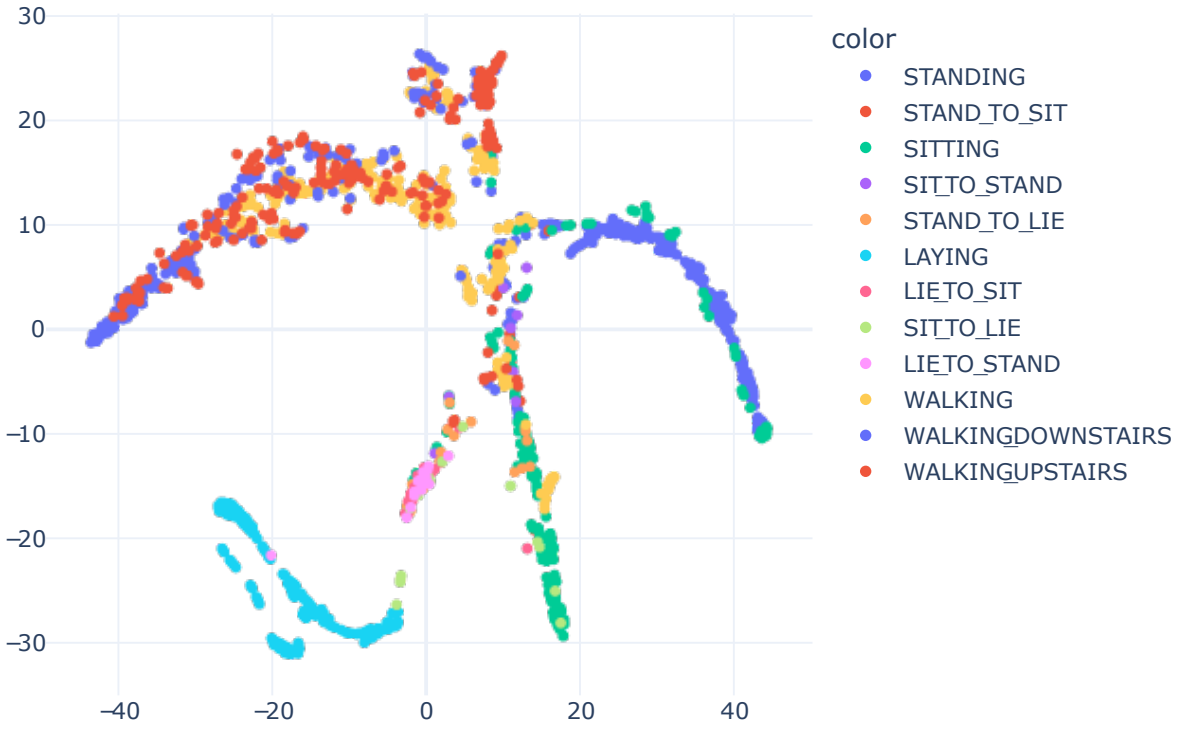


Figure 12. T-SNE Embedding Plot obtained from the CNN1 architecture trained with SI approach with SHO dataset into HAPT dataset. We utilized 40 perplexity and conducted 500 iterations.

6.4. HAPT Features into SHO

To analyze the relevance of the features, we used a network trained on the SHO dataset to extract features from the HAPT dataset, as displayed in Figure 13. To do this, we propagated the HAPT samples through the SHO model and considered the output before the classification layer. We then used this output as input to the t-SNE algorithm.

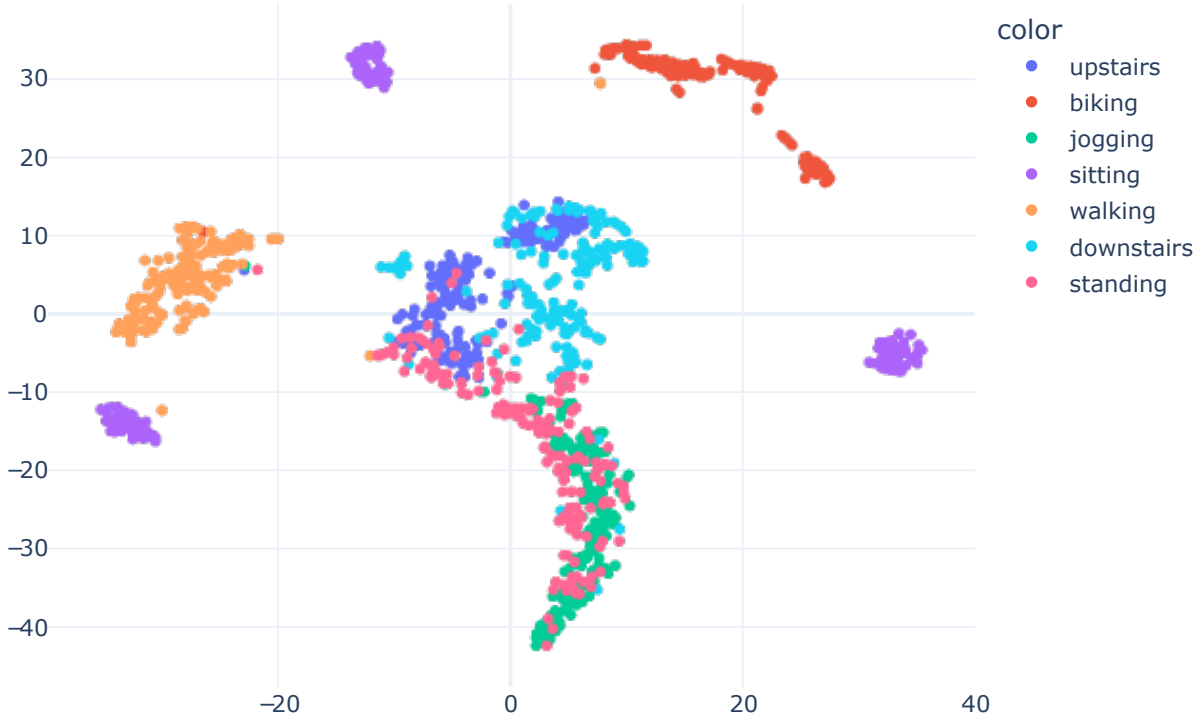


Figure 13. T-SNE Embedding Plot obtained from the CNN1 architecture trained with SI approach with HAPT dataset into SHO dataset. We utilized 40 perplexity and conducted 500 iterations.

When examining the HAPT dataset, it became evident that utilizing learned features resulted in a more precise differentiation of the data than using the raw accelerometer data. Figure 11 shows that the contrast between the two approaches was clear. The Sitting, Walking, and Biking classes were well separated, unlike what was observed when analyzing the raw data dispersion, shown in Figure 8, where the Biking class was grouped with other classes. The Upstairs, Downstairs, Standing, and Jogging classes were grouped in close regions, but there was still noticeable separation between the samples of these classes.

7. Conclusions

T-SNE is frequently utilized to explore the separation of raw data in a comprehensible manner. By applying this method to the output of a DL model, we introduce a novel post-hoc and model-specific approach to the XAI field. Implementing t-SNE on the output of a DL model generates a visualization that can convey a general understanding of the model's learned features during the training process. This is distinct from the explanations produced by other established XAI methods, such as decision tree induction, rule extraction, or gradient-based techniques. In contrast to these alternatives, the proposed methodology is adept at presenting a holistic overview of a DL model across a data subset. By enhancing our understanding of the model's behavior, this technique can function as a valuable instrument for debugging and optimizing the model's performance.

The t-SNE embedding visualization demonstrated its potential in offering valuable insights. Nonetheless, certain limitations must be acknowledged when interpreting the results of this study. For instance, the performance of this visualization technique with alternative data types, such as electrocardiogram signals or other sensor data, remains to be explored, potentially impacting the generalizability of the approach across different domains.

Moreover, although the visualization proved effective in analyzing the decision-making processes of a CNN-based model on identical or diverse datasets, additional research is warranted to assess its applicability to various DL algorithms. Investigating alternative network architectures, such as recurrent neural networks, ConvLSTM, Bidirec-

tional networks, and hybrid nets, may furnish a more comprehensive understanding of the t-SNE embedding visualization's versatility with respect to different deep learning models.

Hence, while the present study contributes valuable insights into the prospective utilization of t-SNE embedding visualization for human activity recognition, based on accelerometer data, it is imperative to recognize the limitations of the proposed approach and continue investigating its efficacy across other domains and with different deep learning algorithms.

Upon applying t-SNE to a network trained on a distinct dataset, the resulting data separation appears less distinct. Instead, the classes display a degree of mixing, as evidenced in Figure 12, where the distribution was inferior to that observed when utilizing raw data as input, as depicted in Figure 11. This confusion may arise from the network's inability to learn pertinent features beyond its original database. Such limitations could stem from discrepancies in data collection, including differences in the individuals involved, the sensors employed, or the collection software implementations. For instance, although both databases position the sensor at the waist using a belt, variations in orientation and characteristics may still arise. If the belt is fastened near the navel, the features obtained may differ from those acquired when the phone is attached closer to the side. The closer the sensor is to the center of mass, the smoother the signal, and this property may influence the neural network's capacity to extract relevant features.

The features extracted by the model trained on the HAPT dataset proved to be more informative, enabling more accurate discrimination of the data than the raw data itself when applied to the SHO dataset, as can be observed when comparing Figures 11 and 13.

The proposed visualization offers a lucid representation of the data distribution based on the network's learned features. This aids in accomplishing several objectives, such as detecting and rectifying critical confusion, pinpointing biases, identifying mislabeled samples, revealing potential subgroups within a group, and more.

This approach is not confined to HAR alone. It can be employed in any problem involving time series data and deep learning algorithms, particularly convolutional neural networks.

This study paves the way for various research avenues, including the following:

1. Which validation strategy results in better separation of classes in a HAR dataset, SI or SD?
2. If additional sensor position data were incorporated into the SHO database, would it improve the separation of HAPT classes?
3. Is it possible to reverse the process? Starting from t-SNE embedding coordinates, can raw data be retrieved? If so, could this be utilized for data augmentation in the dataset?
4. How do the embeddings derived from other deep network models, such as LSTM, BiLSTM, GRU, MLP, and hybrid models, perform?
5. How should the proposed visualization technique be employed to avoid class confusions, modifying the architecture to improve and facilitate the differentiation of challenging classes?

Author Contributions: Conceptualization, G.A., C.F.F.C.F. and M.G.F.C.; methodology, G.A., C.F.F.C.F. and M.G.F.C.; software, G.A.; validation, G.A.; formal analysis, G.A., C.F.F.C.F. and M.G.F.C.; investigation, G.A.; resources, C.F.F.C.F. and M.G.F.C.; data curation, G.A.; writing—original draft preparation, G.A.; writing—review and editing, G.A., C.F.F.C.F. and M.G.F.C.; visualization, G.A.; supervision, C.F.F.C.F. and M.G.F.C.; project administration, C.F.F.C.F. and M.G.F.C.; funding acquisition, M.G.F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 48 of Decree n°6.008/2006 (SUFRAMA), and was funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law n°8.387/1991 through agreement 001/2020, signed with UFAM and FAEPI, Brazil. The authors would also like to thank CAPES for its financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The HAPT dataset can be found on https://archive.ics.uci.edu/ml/datasets/smartphone-based+recognition+of+human+activities+and+postural+transitions (accessed on 11 February 2023). The SHO dataset can be found in the author's ResearchGate profile or https://www.researchgate.net/profile/Muhammad_Shoaib20/publication/266384007_Sensors_Activity_Recognition_DataSet/data/542e9d260cf277d58e8ec40c/Sensors-Activity-Recognition-DataSet-Shoaib.rar (accessed on 11 February 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hayat, A.; Dias, M.; Bhuyan, B.P.; Tomar, R. Human Activity Recognition for Elderly People Using Machine and Deep Learning Approaches. *Informatics* **2022**, *13*, 275. [CrossRef]
2. Gupta, N.; Gupta, S.K.; Pathak, R.K.; Jain, V.; Rashidi, P.; Suri, J.S. Human activity recognition in artificial intelligence framework: A narrative review. *Artif. Intell. Rev.* **2022**, *55*, 4755–4808. [CrossRef] [PubMed]
3. Ferrari, A.; Micucci, D.; Mobilio, M.; Napoletano, P. Trends in human activity recognition using smartphones. *J. Reliab. Intell. Environ.* **2021**, *7*, 189–213. [CrossRef]
4. Zhang, S.; Li, Y.; Zhang, S.; Shahabi, F.; Xia, S.; Deng, Y.; Alshurafa, N. Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances. *Sensors* **2022**, *22*, 1476. [CrossRef]
5. Kazemimoghadam, M.; Fey, N.P. An Activity Recognition Framework for Continuous Monitoring of Non-Steady-State Locomotion of Individuals with Parkinson's Disease. *Appl. Sci.* **2022**, *12*, 4682. [CrossRef]
6. Cvetković, B.; Janko, V.; Romero, A.E.; Kafalı, Ö; Stathis, K.; Luštrek, M. Activity Recognition for Diabetic Patients Using a Smartphone. *J. Med. Syst.* **2016**, *40*, 256. [CrossRef]
7. Gu, F.; Chung, M.H.; Chignell, M.; Valaee, S.; Zhou, B.; Liu, X. A Survey on Deep Learning for Human Activity Recognition. *ACM Comput. Surv.* **2021**, *54*, 3472290. [CrossRef]
8. Aquino, G.; Costa, M.G.; Filho, C.F.C. Explaining One-Dimensional Convolutional Models in Human Activity Recognition and Biometric Identification Tasks. *Sensors* **2022**, *22*, 5644. [CrossRef] [PubMed]
9. Chen, K.; Zhang, D.; Yao, L.; Guo, B.; Yu, Z.; Liu, Y. Deep Learning for Sensor-Based Human Activity Recognition: Overview, Challenges, and Opportunities. *ACM Comput. Surv.* **2021**, *54*, 3447744. [CrossRef]
10. Caldwell, M.; Andrews, J.T.; Tanay, T.; Griffin, L.D. AI-enabled future crime. *Crime Sci.* **2020**, *9*, 14. [CrossRef]
11. Gohel, P.; Singh, P.; Mohanty, M. Explainable AI: Current status and future directions. *arXiv* **2021**, arXiv:2107.07045.
12. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2019**, *128*, 336–359. [CrossRef]
13. Cai, T.T.; Ma, R. Theoretical Foundations of t-SNE for Visualizing High-Dimensional Clustered Data. *arXiv* **2022**, arXiv:2105.07536.
14. van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
15. Rogovschi, N.; Kitazono, J.; Grozavu, N.; Omori, T.; Ozawa, S. t-Distributed stochastic neighbor embedding spectral clustering. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1628–1632. [CrossRef]
16. Bragança, H.; Colonna, J.G.; Oliveira, H.A.; Souto, E. How validation methodology influences human activity recognition mobile systems. *Sensors* **2022**, *22*, 2360. [CrossRef]
17. Micucci, D.; Mobilio, M.; Napoletano, P. UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones. *Appl. Sci.* **2017**, *7*, 101. [CrossRef]
18. Reyes-Ortiz, J.L.; Oneto, L.; Samà, A.; Parra, X.; Anguita, D. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* **2016**, *171*, 754–767. [CrossRef]
19. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H.; Havinga, P.J. A survey of online activity recognition using mobile phones. *Sensors* **2015**, *15*, 2059–2085. [CrossRef]
20. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
21. Samek, W.; Wiegand, T.; Müller, K.R. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. *arXiv* **2017**, arXiv:1708.08296.
22. Datta, A.; Sen, S.; Zick, Y. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 598–617. [CrossRef]
23. Lipovetsky, S.; Conklin, M. Analysis of regression in game theory approach. *Appl. Stoch. Model. Bus. Ind.* **2001**, *17*, 319–330. [CrossRef]

24. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv* **2016**, arXiv:cs.LG/1602.04938.
25. Lu, Z.; Kazi, R.H.; Wei, L.Y.; Dontcheva, M.; Karahalios, K. StreamSketch: Exploring Multi-Modal Interactions in Creative Live Streams. *Proc. ACM Hum.-Comput. Interact.* **2021**, *5*, 1122456. [CrossRef]
26. Gilpin, L.H.; Bau, D.; Yuan, B.Z.; Bajwa, A.; Specter, M.A.; Kagal, L. Explaining Explanations: An Overview of Interpretability of Machine Learning. In Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA) Turin, Italy, 1–3 October 2018; pp. 80–89.
27. Zhang, Q.; Wu, Y.N.; Zhu, S.C. Interpretable Convolutional Neural Networks. *arXiv* **2018**, arXiv:cs.CV/1710.00935.
28. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H.; Havinga, P.J. Fusion of smartphone motion sensors for physical activity recognition. *Sensors* **2014**, *14*, 10146–10176. [CrossRef] [PubMed]
29. Sani, S.; Massie, S.; Wiratunga, N.; Cooper, K. Learning Deep and Shallow Features for Human Activity Recognition. In Proceedings of the Knowledge Science, Engineering and Management, Melbourne, VIC, Australia, 19–20 August 2017; pp. 469–482.
30. Dong, M.; Han, J. HAR-Net:Fusing Deep Representation and Hand-crafted Features for Human Activity Recognition. *arXiv* **2018**, arXiv:cs.LG/1810.10929
31. Juefei-Xu, F.; Bhagavatula, C.; Jaech, A.; Prasad, U.; Savvides, M. Gait-ID on the move: Pace independent human identification using cell phone accelerometer dynamics. In Proceedings of the 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, USA, 23–27 September 2012; pp. 8–15. [CrossRef]
32. Ronao, C.A.; Cho, S.B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.* **2016**, *59*, 235–244. [CrossRef]
33. Thu, N.T.H.; Han, D.S. Utilization of Postural Transitions in Sensor-based Human Activity Recognition. In Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 19–21 February 2020; pp. 177–181. [CrossRef]
34. Thu, N.T.H.; Han, D.S. HiHAR: A Hierarchical Hybrid Deep Learning Architecture for Wearable Sensor-Based Human Activity Recognition. *IEEE Access* **2021**, *9*, 145271–145281. [CrossRef]
35. Jiang, W.; Yin, Z. Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks. In Proceedings of the 23rd ACM International Conference on Multimedia, New York, NY, USA, 20–24 October 2015; pp. 1307–1310. [CrossRef]
36. Viton, F.; Elbattah, M.; Guérin, J.L.; Dequen, G. Heatmaps for Visual Explainability of CNN-Based Predictions for Multivariate Time Series with Application to Healthcare. In Proceedings of the 2020 IEEE International Conference on Healthcare Informatics (ICHI), Oldenburg, Germany, 30 November–3 December 2020; pp. 1–8. [CrossRef]
37. Schlegel, U.; Arnout, H.; El-Assady, M.; Oelke, D.; Keim, D.A. Towards a Rigorous Evaluation of XAI Methods on Time Series. *arXiv* **2019**, arXiv:cs.LG/1909.07082.
38. Schlegel, U.; Keim, D.A. Time Series Model Attribution Visualizations as Explanations. In Proceedings of the 2021 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX), New Orleans, LA, USA, 24–25 October 2021.
39. Assaf, R.; Schumann, A. Explainable Deep Neural Networks for Multivariate Time Series Predictions. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, Macao, China, 10–16 August 2019; pp. 6488–6490. [CrossRef]
40. Suzuki, S.; Amemiya, Y.; Sato, M. Skeleton-based explainable human activity recognition for child gross-motor assessment. In Proceedings of the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 4015–4022. [CrossRef]
41. Riboni, D.; Bettini, C.; Civitarese, G.; Janjua, Z.H.; Helaoui, R. SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment. *Artif. Intell. Med.* **2016**, *67*, 57–74. [CrossRef] [PubMed]
42. Arrotta, L.; Civitarese, G.; Bettini, C. Dexar: Deep explainable sensor-based activity recognition in smart-home environments. *Proc. Acm Interact. Mob. Wearable Ubiquitous Technol.* **2022**, *6*, 1–30. [CrossRef]
43. Bettini, C.; Civitarese, G.; Fiori, M. Explainable Activity Recognition over Interpretable Models. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), Kassel, Germany, 22–26 March 2021; pp. 32–37. [CrossRef]
44. Atzmueller, M.; Hayat, N.; Trojahn, M.; Kroll, D. Explicative human activity recognition using adaptive association rule-based classification. In Proceedings of the 2018 IEEE International Conference on Future IoT Technologies (Future IoT), Eger, Hungary, 18–19 January 2018; pp. 1–6. [CrossRef]
45. Dharavath, R.; MadhukarRao, G.; Khurana, H.; Edla, D.R. t-SNE Manifold Learning Based Visualization: A Human Activity Recognition Approach. In Proceedings of the Advances in Data Science and Management, Changsha, China, 20–21 September 2020; pp. 33–43.
46. Thakur, D.; Guzzo, A.; Fortino, G. t-SNE and PCA in Ensemble Learning based Human Activity Recognition with Smartwatch. In Proceedings of the 2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS), Magdeburg, Germany, 8–10 September 2021; pp. 1–6. [CrossRef]

47. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
48. Wan, S.; Qi, L.; Xu, X.; Tong, C.; Gu, Z. Deep Learning Models for Real-time Human Activity Recognition with Smartphones. *Mob. Netw. Appl.* **2020**, *25*, 743–755. [CrossRef]

Referências Bibliográficas

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y. & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. Em *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, p. 265–283, USA. USENIX Association.

Adebayo, J.; Muelly, M.; Liccardi, I. & Kim, B. (2020). Debugging tests for model explanations.

Aquino, G.; Costa, M. G. & Filho, C. F. C. (2022). Explaining one-dimensional convolutional models in human activity recognition and biometric identification tasks. *Sensors*, 22.

Aquino, G.; Costa, M. G. F. & Filho, C. F. F. C. (2023). Explaining and visualizing embeddings of one-dimensional convolutional models in human activity recognition tasks. *Sensors*, 23(9).

Arrotta, L.; Civitarese, G. & Bettini, C. (2022). Dexar: Deep explainable sensor-based activity recognition in smart-home environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6.

Assaf, R. & Schumann, A. (2019). Explainable deep neural networks for multivariate time series predictions. Em *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 6488--6490. International Joint Conferences on Artificial Intelligence Organization.

Atzmueller, M.; Hayat, N.; Trojahn, M. & Kroll, D. (2018). Explicative human activity recognition using adaptive association rule-based classification. Em *2018 IEEE International Conference on Future IoT Technologies (Future IoT)*, pp. 1–6.

Bettini, C.; Civitarese, G. & Fiori, M. (2021). Explainable activity recognition over interpretable models. Em *2021 IEEE International Conference on Pervasive*

Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp. 32–37.

Bishop, C. (2016). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York.

Booth, F. W.; Roberts, C. K. & Laye, M. J. (2012). Lack of exercise is a major cause of chronic diseases. *Comprehensive physiology*, 2(2):1143.

Bragança, H.; Colonna, J. G.; Oliveira, H. A. B. F. & Souto, E. (2022). How validation methodology influences human activity recognition mobile systems. *Sensors*, 22:2360.

Caldwell, M.; Andrews, J. T.; Tanay, T. & Griffin, L. D. (2020). Ai-enabled future crime. *Crime Science*, 9.

Chen, K.; Zhang, D.; Yao, L.; Guo, B.; Yu, Z. & Liu, Y. (2021). Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Comput. Surv.*, 54(4).

Cheng, X.; Zhang, L.; Tang, Y.; Liu, Y.; Wu, H. & He, J. (2022). Real-time human activity recognition using conditionally parametrized convolutions on mobile and wearable devices. *IEEE Sensors Journal*, 22(6):5889--5901.

Costa, M. G. F.; E Aquino, G. D. A.; Serrão, M. K.; Vieira, D. G. D. A.; De Alencar, E. D. N.; De Negreiro, J. V.; Da Silva, E. S. & Costa Filho, C. F. F. (2021). A new approach for detecting activities of daily living detection using smartphone and wearable sensors. Em *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 01–07.

Costa Filho, C. F.; e Aquino, G. d. A. & Costa, M. G. F. (2021). Gait analysis: determining heel-strike and toe-off events. Em *17th International Symposium on Medical Information Processing and Analysis*, volume 12088, pp. 19--26. SPIE.

Cvetković, B.; Janko, V.; Romero, A. E.; Özgür Kafalı; Stathis, K. & Luštrek, M. (2016). Activity recognition for diabetic patients using a smartphone. *Journal of Medical Systems*, 40.

Datta, A.; Sen, S. & Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. Em *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 598–617.

de Aquino e Aquino, G.; Serrão, M.; Costa, M. & Costa-Filho, C. (2020). Human activity recognition from accelerometer data with convolutional neural networks. Em *Brazilian Congress on Biomedical Engineering*, pp. 1603--1610. Springer.

der Maaten, L. V. & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579--2605.

Ferrari, A.; Micucci, D.; Mobilio, M. & Napoletano, P. (2021). Trends in human activity recognition using smartphones. *Journal of Reliable Intelligent Environments*, 7:189–213.

Foerster, F.; Smeja, M. & Fahrenberg, J. (1999). Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior*, 15(5):571–583.

Gamble, J. A. & Huang, J. (2020). Convolutional neural network for human activity recognition and identification. Em *2020 IEEE International Systems Conference (SysCon)*, pp. 1--7. IEEE.

Gilpin, L. H.; Bau, D.; Yuan, B. Z.; Bajwa, A.; Specter, M. A. & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89.

Glorot, X.; Bordes, A. & Bengio, Y. (2011). Deep sparse rectifier neural networks. Em *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315--323. JMLR Workshop and Conference Proceedings.

Gohel, P.; Singh, P. & Mohanty, M. (2021). Explainable AI: current status and future directions. *CoRR*, abs/2107.07045.

Goodfellow, I.; Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Gu, F.; Chung, M.-H.; Chignell, M.; Valaee, S.; Zhou, B. & Liu, X. (2021). A survey on deep learning for human activity recognition. *ACM Comput. Surv.*, 54(8).

Gupta, N.; Gupta, S. K.; Pathak, R. K.; Jain, V.; Rashidi, P. & Suri, J. S. (2022). Human activity recognition in artificial intelligence framework: a narrative review. *Artificial Intelligence Review*, 55:4755–4808.

Hayat, A.; Dias, M.; Bhuyan, B. P. & Tomar, R. (2022). Human activity recognition for elderly people using machine and deep learning approaches. *Information (Switzerland)*, 13.

Hedström, A.; Weber, L.; Bareeva, D.; Krakowczyk, D.; Motzkus, F.; Samek, W.; Lapuschkin, S. & Höhne, M. M. C. (2023). Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M. & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications.

Jiang, W. & Yin, Z. (2015). Human activity recognition using wearable sensors by deep convolutional neural networks. Em *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, p. 1307–1310, New York, NY, USA. Association for Computing Machinery.

Jolliffe, I. T. & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

Juefei-Xu, F.; Bhagavatula, C.; Jaech, A.; Prasad, U. & Savvides, M. (2012). Gait-id on the move: Pace independent human identification using cell phone accelerometer dynamics. Em *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 8–15.

Kazemimoghadam, M. & Fey, N. P. (2022). An activity recognition framework for continuous monitoring of non-steady-state locomotion of individuals with parkinson's disease. *Applied Sciences (Switzerland)*, 12.

Krizhevsky, A.; Sutskever, I. & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84--90.

LeCun, Y.; Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, 521(7553):436--444.

LeCun, Y.; Bottou, L.; Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278--2324.

Lipovetsky, S. & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17:319–330.

Lu, Z.; Kazi, R. H.; Wei, L. Y.; Dontcheva, M. & Karahalios, K. (2021). Streamsketch: Exploring multi-modal interactions in creative live streams. *Proceedings of the ACM on Human-Computer Interaction*, 5.

Lv, T.; Wang, X.; Jin, L.; Xiao, Y. & Song, M. (2020). A hybrid network based on dense connection and weighted feature aggregation for human activity recognition. *IEEE Access*, 8:68320--68332.

Mekruksavanich, S. & Jitpattanakul, A. (2021). Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models. *Electronics*, 10(3):308.

Micucci, D.; Mobilio, M. & Napoletano, P. (2017). Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10).

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill.

Mukherjee, D.; Mondal, R.; Singh, P. K.; Sarkar, R. & Bhattacharjee, D. (2020). Ensemconvnet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications. *Multimedia Tools and Applications*, 79:31663--31690.

Palatnik de Sousa, I.; Vellasco, M. M. & Costa da Silva, E. (2021). Explainable artificial intelligence for bias detection in covid ct-scan classifiers. *Sensors*, 21(16):5657.

Research, M. (2022). Wearable technology market by product (smartwatches, blood pressure monitor watches, head-mounted displays, smart headgears, smart glasses, smart jewellery, body-worn cameras), material, end user, and geography - global forecast to 2029. Available online.

Reyes-Ortiz, J.-L.; Oneto, L.; Samà, A.; Parra, X. & Anguita, D. (2016). Transition-aware human activity recognition using smartphones. *Neurocomput.*, 171:754–767.

Ribeiro, M. T.; Singh, S. & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier.

Rodgers, J. L. & Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *American statistician*, pp. 59--66.

Ronao, C. A. & Cho, S.-B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, 59:235–244.

Rumelhart, D. E.; Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533--536.

Russell, S. J. & Norvig, P. (2010). *Artificial intelligence: a modern approach*. Pearson Education.

Samek, W.; Wiegand, T. & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *CoRR*, abs/1708.08296.

Sani, S.; Massie, S.; Wiratunga, N. & Cooper, K. (2017). Learning deep and shallow features for human activity recognition. Em Li, G.; Ge, Y.; Zhang, Z.; Jin, Z. & Blumenstein, M., editores, *Knowledge Science, Engineering and Management*, pp. 469--482, Cham. Springer International Publishing.

Schlegel, U.; Arnout, H.; El-Assady, M.; Oelke, D. & Keim, D. A. (2019). Towards a rigorous evaluation of xai methods on time series.

Schlegel, U. & Keim, D. A. (2021). Time series model attribution visualizations as explanations. *CoRR*, abs/2109.12935.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D. & Batra, D. (2019). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336--359.

Serrão, M.; de A. e Aquino, G.; Costa, M. & Costa Filho, C. F. F. (2021). Human activity recognition from accelerometer with convolutional and recurrent neural networks. *Polytechnica*, 4:15--25.

Shoaib, M.; Bosch, S.; Incel, O. D.; Scholten, H. & Havinga, P. J. (2014). Fusion of smartphone motion sensors for physical activity recognition. *Sensors (Switzerland)*, 14:10146–10176.

Shoaib, M.; Bosch, S.; Incel, O. D.; Scholten, H. & Havinga, P. J. (2015). A survey of online activity recognition using mobile phones. *Sensors (Switzerland)*, 15:2059–2085.

Sikder, N.; Chowdhury, M. S.; Arif, A. S. M. & Nahid, A.-A. (2019). Human activity recognition using multichannel convolutional neural network. Em *2019 5th International conference on advances in electrical engineering (ICAEE)*, pp. 560--565. IEEE.

Simonyan, K. & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958.

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tang, Y.; Teng, Q.; Zhang, L.; Min, F. & He, J. (2020). Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors. *IEEE Sensors Journal*, 21(1):581--592.

Teng, Q.; Zhang, L.; Tang, Y.; Song, S.; Wang, X. & He, J. (2021). Block-wise training residual networks on multi-channel time series for human activity recognition. *IEEE Sensors Journal*, 21(16):18063--18074.

Thu, N. T. H. & Han, D. S. (2020). Utilization of postural transitions in sensor-based human activity recognition. Em *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 177–181.

Thu, N. T. H. & Han, D. S. (2021). Hihar: A hierarchical hybrid deep learning architecture for wearable sensor-based human activity recognition. *IEEE Access*, 9:145271–145281.

Viton, F.; Elbattah, M.; Guérin, J.-L. & Dequen, G. (2020). Heatmaps for visual explainability of cnn-based predictions for multivariate time series with application to healthcare. Em *2020 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 1–8.

Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern recognition*, 48(9):2839--2846.

Zhang, Q.; Wu, Y. N. & Zhu, S.-C. (2018). Interpretable convolutional neural networks.

Zhang, S.; Li, Y.; Zhang, S.; Shahabi, F.; Xia, S.; Deng, Y. & Alshurafa, N. (2022). Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors*, 22:1476. HAR systems enables:
activity tracklin, wellness monitoring and human-computer interaction.

Goal: categorizing and sumarizes existing works about HAR.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A. & Torralba, A. (2016). Learning deep features for discriminative localization. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921--2929.