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

COMPRESSÃO DE MODELOS DE RECONHECIMENTO DE ATIVIDADES HUMANAS USANDO DESTILAÇÃO DE CONHECIMENTO

Manaus - AM

Maio de 2022

PAULO HENRIQUE NELLESSEN GONÇALVES

COMPRESSÃO DE MODELOS DE RECONHECIMENTO DE ATIVIDADES HUMANAS USANDO DESTILAÇÃO DE CONHECIMENTO

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito para a obtenção do grau de Mestre em Informática.

ORIENTADOR: PROF. DR. EDUARDO JAMES PEREIRA SOUTO

Manaus - AM

Maio de 2022

Ficha Catalográfica

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

Gonçalves, Paulo Henrique Nellessen
G643c Compressão de modelos de reconhecimento de atividades humanas usando destilação de conhecimento / Paulo Henrique Nellessen Gonçalves . 2022
56 f.: il. color; 31 cm.

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

1. reconhecimento de atividades humanas. 2. smartphone. 3. redes neurais profundas. 4. destilação de conhecimento. 5. compressão. I. Souto, Eduardo Pereira James. II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"Compressão de Modelos de Reconhecimento de Atividades Humanas Usando Destilação de Conhecimento"

PAULO HENRIQUE NELLESSEN GONÇALVES

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Eduardo James Pereira Souto - PRESIDENTE

Prof. Rafael Giusti - MEMBRO INTERNO

Dr. Wesllen Sousa Lima - MEMBRO EXTERNO

Manaus, 17 de Maio de 2022

Dedico este trabalho à minha mãe, ela não só meu deu o dom da vida como me ensinou o que é amar.

AGRADECIMENTOS

Agradeço a minha mãe, Sheila Regina, e ao meu pai, Paulo César, que sempre me deram suporte para estudar e me apoiaram nas minhas decisões acadêmicas. Agradeço também a minha companheira Ayala Bernardo que esteve nessa jornada e me deu suporte nos altos e principalmente nos baixos deste caminho. Agradeço aos meus amigos que não contribuíram diretamente com esta pesquisa, mas fazem parte do meu refúgio espiritual.

Agradeço ao meu orientador professor Dr. Eduardo Souto pela paciência, pelo profissionalismo, dedicação e orientação em pontos importantes da minha pesquisa. Agradeço ao meus amigos de laboratório que demostraram companheirismo e auxiliaram o desenvolvimento do meu projeto compartilhando conhecimento e risadas. Agradeço especialmente meu amigo Hendrio Bragança pelas sugestões e auxílio em vários pontos importantes para conclusão desta pesquisa.

Agradeço a CAPES pelo suporte financeiro essencial, à UFAM pelo por proporcionar um ambiente bem equipado para o desenvolvimento desta pesquisa e a todos os professores que contribuíram para minha formação acadêmica.

"Oh take your time
don't live too fast
Troubles will come
and they will pass."
Van Zant Ronnie / Rossington Gary

RESUMO

O uso de dispositivos móveis e vestíveis tem possibilitado o monitoramento contínuo das atividades realizadas pelo usuário. Entretanto, esse processo é desafiador devido à natureza complexa dos dados capturados pelos sensores disponíveis nestes dispositivos. Recentemente, o uso de redes neurais profundas tem ampliado os limites para reconhecer atividades humanas com alta precisão. No entanto, no contexto móvel e vestível, as restrições de *hardware* podem inviabilizar o uso de redes neurais profundas, pois os recursos computacionais são limitados. Para mitigar as limitações relacionadas ao custo computacional de redes neurais profundas, este trabalho propõe um método chamado KD-HAR (*Knowledge Distillation for Human Activity Recognition*) para compressão de redes neurais profundas baseado na técnica de destilação de conhecimento aplicada a modelos de reconhecimento de atividade humana usando dados de sensores inerciais. Os conhecimentos adquiridos por modelos professores, obtidos por meio de técnicas de otimização de hiperparâmetros, são transferidos para modelos estudantes com menor complexidade. Uma das vantagens do método proposto é a capacidade de extração automática de características, baixo custo computacional e a aproximação da precisão na classificação das atividades quando comparado a redes mais complexas. Para avaliar a capacidade de compressão do método proposto, este trabalho utiliza duas bases de dados (UCI-HAR e WISDM) de sensores inerciais de *smartphones*. Os resultados obtidos mostram que o método é capaz de manter acurácia competitiva com taxas de compressão que variam de 18 a 42 vezes o número de parâmetros da rede neural profunda destilada em relação ao modelo de professor treinado.

Palavras-chave: reconhecimento de atividades humanas, *smartphone*, redes neurais profundas, destilação de conhecimento, compressão.

ABSTRACT

The use of mobile and wearable devices can enable continuous monitoring of activities performed by the user. However, this process is challenging due to the complex nature of data devices by available devices. Recently, the use of deep neural networks has pushed the limits to recognize human activities with high accuracy. However, in the wearable mobile context, hardware restriction can make the use of deep neural networks unfeasible, as computational resources are limited. To mitigate as related to the computational cost of deep neural networks, this project work of a method called KD-HAR (*Knowledge Distillation for Human Activity Recognition*) for deep neural network compression based on the knowledge distillation technique applied to models of human activity recognition using data from inertial sensors. The knowledge acquired by teacher models, through hyperparameter optimization techniques, are transferred to student models with less complexity. One of the advantages of the proposed method is the ability to automatically extract features, low computational cost and the approximation of precision in the classification of activities when compared to more complex networks. To evaluate the compression capacity of the proposed method, this work uses two databases (UCI-HAR and WISDM) of smartphone inertial sensors. The results obtained show that the method can maintain competitive accuracy with compression rates ranging from 18 to 42 times the number of parameters of the distilled deep neural network in relation to the trained teacher model.

Keywords: human activity recognition, smartphone, deep neural networks, knowledge distillation, compression.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Na última década houve um desenvolvimento exponencial de microeletrônicos e sistemas computacionais, o que viabilizou a criação de sensores embarcados e dispositivos móveis com alto poder computacional, tamanho pequeno e baixo custo (LARA; LABRADOR, 2013; CHEN *et al.*, 2021). A ampla adoção desses dispositivos é evidente no dia a dia da população, especialmente sobre o uso de *smartphones* e mais atualmente, com a popularização de *smartbands* e *smartwatches*, usados por uma parcela representativa da população. Esses dispositivos são equipados com diversos sensores, tais como acelerômetro, giroscópio e magnetômetro, tornando-os assim, adequados para monitoramento automático de diversos aspectos relacionados ao cotidiano de seus usuários. Os dados valiosos obtidos por meio desse monitoramento podem ser usados para prover serviços personalizados aos seus usuários (WANG *et al.*, 2019b).

Através de informações sobre a rotina de um indivíduo, é possível monitorar e/ou auxiliar a execução de suas atividades em diversos domínios. Por exemplo, monitorar sinais vitais para gerar alertas quando houver anomalias (LEE; JEONG; YOON, 2012), realizar a contagem de passos e detecção de caminhada para monitoramento esportivo (BRAJDIC; HARLE, 2013), fazer a detecção de quedas visando evitar situações de risco de vida (BOEHNER, 2013) e controlar e garantir a segurança em casas inteligentes para melhorar a experiência do usuário (WILSON; HARGREAVES; HAUXWELL-BALDWIN, 2015).

Nesse contexto, o Reconhecimento de Atividades Humanas (RAH) é uma área de pesquisa cujo objetivo principal é reconhecer atividades físicas desempenhadas por um indivíduo (YANG; LEE; CHOI, 2011; LARA; LABRADOR, 2013). O RAH com dados de sensores de *smartphone* se mostrou recentemente uma das melhores estratégias para monitorar a execuções de atividades por sua natureza não intrusiva e contínua. O RAH vai desde o reconhecimento de um conjunto comum de ações como, por exemplo, andar e correr até mais complexas como cozinhar ou preparar café.

Quando essas ações são executadas, os sensores capturam as variações ao longo do tempo. Portanto, é necessário capturar os padrões destas variações de forma que cada padrão determine uma ação especifica. Nesse contexto, os algoritmos de aprendizado de máquina (do inglês - *Machine Learning* - ML) podem ser usados para capturar características dessas ações, ou atividades, e classificá-las de forma correta.

Uma das abordagens mais comuns para se classificar uma atividade com dados capturados de sensores se inicia através da segmentação dos dados e aplicação de funções estatísticas selecionadas de forma manual, as quais buscam capturar características de

tempo (e.g. média e desvio padrão) e frequência (e.g. energia e frequência dominante) (ANGUITA *et al.*, 2013). Os dados resultantes são então passados para um modelo de aprendizado de máquina, responsável pela classificação das atividades executadas.

Portanto, no processo de extração manual é necessário um especialista na área, com domínio sobre o tipo de atividade executada, o qual deve aplicar as funções de domínio de tempo e frequência cuidadosamente selecionadas para que os dados contenham um conjunto de características capazes de proporcionar uma melhor representatividade das atividades, visando uma boa acurácia no modelo de classificação. Por outro lado, visando contornar as dificuldades resultantes da extração manual de características, vários trabalhos buscam automatizar o processo de extração de características para minimizar a intervenção do especialista (NWEKE *et al.*, 2019).

Mais recentemente, algoritmos de aprendizagem de máquina mais robustos e complexos, conhecidos como Redes Neurais Profundas (RNPs), atingiram o desempenho em nível sobrehumano em vários domínios, incluindo RAH (CHEN *et al.*, 2021). Diferente dos modelos tradicionais de ML, as redes neurais profundas têm a capacidade de aprender automaticamente características relevantes dos dados dos sensores, pois todo o processo é realizado dentro das camadas ocultas da rede. Esse processo pode eliminar por completo a necessidade de um especialista dedicado a uma fase intermediária de extração de características. Uma outra vantagem dessas redes é sua superioridade em desempenho quando aplicados a grandes massas de dados, se comparados também aos modelos tradicionais de ML (NWEKE *et al.*, 2019).

As soluções que usam RNPs mais comuns encontradas na literatura são baseadas em *Convolutional Neural Networks* (CNN) e *Long-Short-Term Memory Recurrent* (LSTMs) (RAVI *et al.*, 2016; RONAO; CHO, 2016; IGNATOV, 2018; WANG *et al.*, 2019a). A extração de características dos dados dos sensores é realizada nas camadas de convolução (YANG *et al.*, 2015).Essa combinação de camadas CNN e LSTM é uma das formas eficientes de trabalhar com dados de atividades, sendo capaz de capturar dependência local e variância de escala (Xue *et al.*, 2018).

Infelizmente, uma das principais desvantagens que os algoritmos baseados em redes neurais profundas está relacionada ao seu alto custo computacional, não apenas devido à alta complexidade computacional, mas também aos grandes requisitos de armazenamento, energia e latência, o que pode tornar sua implementação inadequada para a criação de aplicações RAH em tempo real implementadas em dispositivos com baixo poder computacional, como é o caso dos dispositivos *IoT*, *smartbands* e *smartwatches* (RAVI *et al.*, 2016; IGNATOV, 2018).

Para mitigar problemas relacionados às limitações de recursos de dispositivos com baixa capacidade computacional, uma variedade de técnicas de compressão de modelos foi desenvolvida, como por exemplo, poda (SRINIVAS; BABU, 2015), quantização (GONG *et*

al., 2014; WU *et al.*, 2016), e destilação de conhecimento (CHENG *et al.*, 2017).

A destilação do conhecimento, do inglês *Knowledge Distillation* (KD), é uma técnica que transfere conhecimento aprendido por uma RNP, que funciona como uma rede professor, para uma rede leve, ou rede estudante. Essa técnica tem se mostrado eficaz para compactar redes neurais (HINTON *et al.*, 2015; ROMERO *et al.*, 2014; CHENG *et al.*, 2020; GOU *et al.*, 2021; VU; LE; WANG, 2021). Essa forma de transferência de aprendizagem tem recebido atenção da comunidade, principalmente porque viabiliza aplicações desenvolvidas para dispositivos móveis e IoT (*Internet of Things*) (GOU *et al.*, 2021). Este trabalho adota a destilação de conhecimento para comprimir e transferir conhecimento de modelos utilizados como estado da arte na literatura de RAH para modelos mais simples que podem ser implementados em dispositivos com limitações computacionais.

Na destilação do conhecimento, um modelo estudante, mais simples e menor complexidade, é supervisionado por um modelo professor complexo. A ideia principal é que o modelo de estudante imite o modelo professor para obter um desempenho competitivo ou mesmo superior (HINTON *et al.*, 2015; ROMERO *et al.*, 2014; CHENG *et al.*, 2020; GOU *et al.*, 2021). O desafio chave é traçar estratégias de como transferir o conhecimento de um modelo professor para um modelo estudante de baixa complexidade, enquanto mantendo a capacidade de generalização aprendida pelo modelo professor.

Este trabalho propõe um método de compressão de modelos aplicado no contexto de reconhecimento de atividade humanas para treinar redes viáveis para implementação em dispositivos móveis. O método proposto é baseado em três conceitos chaves usados em KD: conhecimento, algoritmo de destilação e arquitetura professor-estudante. Nosso método de compressão de modelos pode ser aplicado no contexto RAH para entregar modelos de baixo custo e desempenho competitivo ou superior aos modelos encontrados no estado da arte. Nosso método implementa um paradigma estudante-professor, no qual o modelo estudante é penalizado de acordo com as predições do modelo professor. Para isso, o estudante é treinado para prever a saída do professor, bem como os rótulos originais encontrados na base de dados (HINTON *et al.*, 2015; ROMERO *et al.*, 2014). Em contrapartida, o modelo estudante apresenta uma arquitetura compacta se comparada ao modelo professor. O modelo estudante obtido como resultado da destilação é compacto e pode ser facilmente implantado em aplicativos do mundo real.

1.1 Objetivos

Desenvolver e avaliar um método para compressão de modelos baseados em redes neurais profundas no contexto de reconhecimento de atividades humanas. O método, denominado KD-HAR (*Knowledge Distillation for Human Activity Recognition*), utiliza a técnica de destilação de conhecimento aplicada a modelos de reconhecimento de atividades humanas cujo objetivo é gerar modelos de HAR embarcáveis em dispositivos com restrição

de recursos de *hardware* e ao mesmo tempo manter o desempenho similar aos modelos estado-da-arte encontrados na literatura. O método KD-HAR emprega técnicas de otimização de hiperparâmetros para geração de um modelo de alto desempenho, complexo e de alta capacidade, denominado professor, e na sequência destila o conhecimento do modelo professor para uma rede compacta, denominada estudante.

1.1.1 Objetivos específicos

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

1. Projetar uma arquitetura de rede neural profunda que será utilizada para gerar modelos professores. O modelo professor candidato é escolhido baseado em técnicas de otimização de hiperparâmetros.
2. Projetar uma arquitetura de rede neral profunda que será utilizada gerar um modelo estudante. O modelo estudando possui uma natureza mais compacta em relação ao modelo professor.
3. Desenvolver um estratégia de treinamento de um conjunto de modelos estudantes aplicando a técnica de destilação de conhecimento.

1.2 Organização do Texto

O restante deste trabalho está organizado da seguinte maneira:

- Capítulo 2 apresenta os conceitos fundamentais para a compreensão dos elementos que compõe o método proposto. Entre os conceitos apresentados estão o reconhecimento de atividades humanos com dados de sensores inerciais, conceitos de Redes Neurais Profundas e a destilação de conhecimento para compressão de Redes Neurais Profundas.
- Capítulo 3 apresenta trabalhos que influenciaram o desenvolvimento desta pesquisa. Os trabalhos são divididos em trabalhos sobre destilação de conhecimento no contexto de reconhecimento de atividades humanas e trabalhos estado-da-arte para o reconhecimento de atividades humanas.
- Capítulo 4 detalha o método proposto denominado KD-HAR (*Knowledge Distillation for Human Activity Recognition*) para a compressão de RNPs baseado na técnica de destilação de conhecimento aplicada à modelos de reconhecimento de atividades humanas usando dados de sensores inerciais.
- Capítulo 5 contém o protocolo experimental utilizado e apresenta os resultados obtidos, englobando as métricas de avaliação, base de dados e método de validação.

O método é testado em duas bases de dados no contexto de reconhecimento de atividades humanas, buscando validar o método proposto em relação a compressão dos modelos de Rede Neural Profunda e a manutenção do desempenho na classificação das atividades avaliadas.

- No capítulo 6 é realizada uma conclusão sobre os resultados encontrados com a aplicação do método e são descritos potenciais trabalhos futuros.

2 RECONHECIMENTO DE ATIVIDADES HUMANAS

Este capítulo aborda os conceitos necessários para o entendimento geral desta pesquisa. Nele é descrito como o reconhecimento de atividade pode ser utilizado através dos dados de sensores inerciais, tais como os embarcados em smartphones. Na sequência, são descritos os dados inerciais e como o pré-processamento é executado. Em seguida, são explicados os conceitos sobre Redes Neurais Profundas (RNP), com foco teórico sobre redes neurais convolutivas. Por fim, os conceitos sobre a destilação de conhecimento para compressão de RNPs são abordados.

2.1 Definição do problema

As aplicações para reconhecimento de atividade em sua grande maioria utilizam dados provenientes dos sensores embutidos, os quais são capturados de forma constante e formam diversas séries temporais. Sendo assim, o reconhecimento está associado à solução do seguinte problema não determinístico (LARA; LABRADOR, 2013).

Dado um conjunto $S = S_0, ..., S_{k-1}$ de k séries temporais, cada uma de um atributo particular medido, e todos definidos dentro do intervalo $I = [t_\alpha, ..., S_\omega]$, o objetivo é encontrar um partição temporal $I = \{I_0, ..., I_{r-1}\rangle$, com base nos dados do conjunto S, e um conjunto de rótulos representando a atividade durante cada intervalo I_j (e.g. sentando, andando, etc.). Essa condição implica que os intervalos de tempo são consecutivos, não vazios, sem sobreposição e dessa forma, $\bigcup_{j=0}^{r-1} I_j = I$

Entretanto, essa definição implica que as atividades devem ser identificadas do instante iniciada até seu fim, ou seja, cada intervalo temporal pode ser variável e o problema associado de se encontrar pontos de transição entre atividades é difícil, visto que as durações das atividades são normalmente desconhecidas. Dessa forma, trabalhos na área costumam utilizar uma definição mais relaxada (LARA; LABRADOR, 2013)

Dado (1) um conjunto $W = W_0, ..., W_{m-1}$ de tamanho fixo m, totalmente ou parcialmente rotulados, de forma que cada W_i contém um conjunto de séries temporais $S = S_{i,0}, ..., S_{i,k-1}$ para cada um dos k atributos mensurados, e (2) um conjunto $A = a_0, ..., a_{n-1}$ de rótulos das atividades, o objetivo é encontrar uma função que mapeie $f : S_i \rightarrow A$, o qual possa ser avaliado para todos valores de S_i, tal que $f(S_i)$ é o mais similar possível da atividade real executada durante W_i.

Na forma relaxada do problema de RAH é importante observar que erros são introduzidos durante a inferência de atividades, pois dada uma instância W_i associada a um conjunto S, o rótulo associado mapeado é único, mas uma pessoa pode executar mais de uma atividade durante um intervalo fixo e portanto, a série temporal S irá conter

características de outras atividades (e.g. um usuário anda durante 4 segundos e para durante outros 2 segundos, nesse intervalo, o rótulo associado é de andando, porém há dados da atividade parado).

Definido o problema, nas próximas seções será explicado o processo de RAH dividido em 5 etapas principais (Figura 1): aquisição de dados (dos sensores), pré-processamento, segmentação, extração de características e classificação (SHOAIB *et al.*, 2015). Esta pesquisa busca explorar técnicas de extração automática, especificamente redes neurais convolutivas. Dessa forma, o processo de extração de características e classificação são unidos e consideramos o processo em 4 etapas, onde a extração de características e classificação é executada por uma RNP. As subseções seguintes irão abordar o processo de RAH até segmentação, de onde irá seguir apenas o caminho sobre redes neurais.

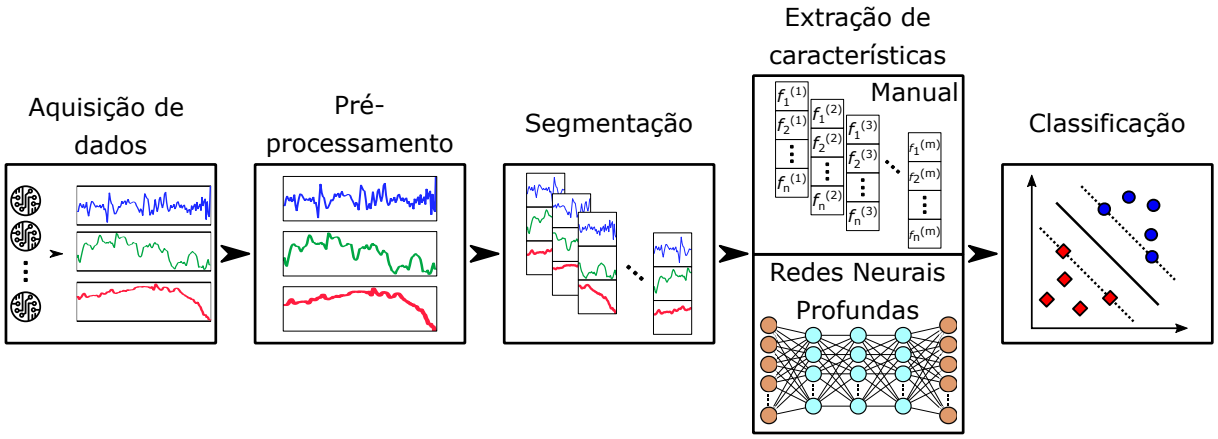


Figura 1 – Processo de reconhecimento de atividades humanas em 5 etapas. Adaptado de Li *et al.* (2018) .

2.2 Aquisição de dados de sensores em smartphones

Existem diversos sensores embutidos em *smartphones* e, a escolha adequada de quais sensores utilizar está diretamente relacionada com a atividade a ser monitorada e sobre as particularidades provenientes do sensor estudado. Atualmente, é comum encontrar nos *smartphones* os seguintes sensores: acelerômetro, giroscópio, magnetômetro, termômetro, sensor de luz, barômetro, microfone, câmera e GPS (SHOAIB *et al.*, 2014). Além disso, dispositivos vestíveis, como os *smartbands* e fones inteligentes, usualmente possuem sensores semelhantes aos encontrados em *smartphones* (com exclusão da câmera).

Apesar da ampla variedade de sensores, esta pesquisa foca no uso de sensores inerciais, tais como o acelerômetro, giroscópio e magnetômetro, pois eles são capazes de monitorar o movimento do *smartphone* e outros dispositivos vestíveis, como vibrações, rotações e oscilações, que normalmente são reflexo direto da interação do usuário sobre o ambiente. Esses sensores fazem capturas constantes em 3 eixos (x, y, z) e retornam um

conjunto $S = (t_i)_{i=1}^{N}, t_i \in R^3$. Por exemplo, a Figura 2 exemplifica dados de acelerômetro triaxial durante a execução de 6 atividades (ANGUITA *et al.*, 2013).

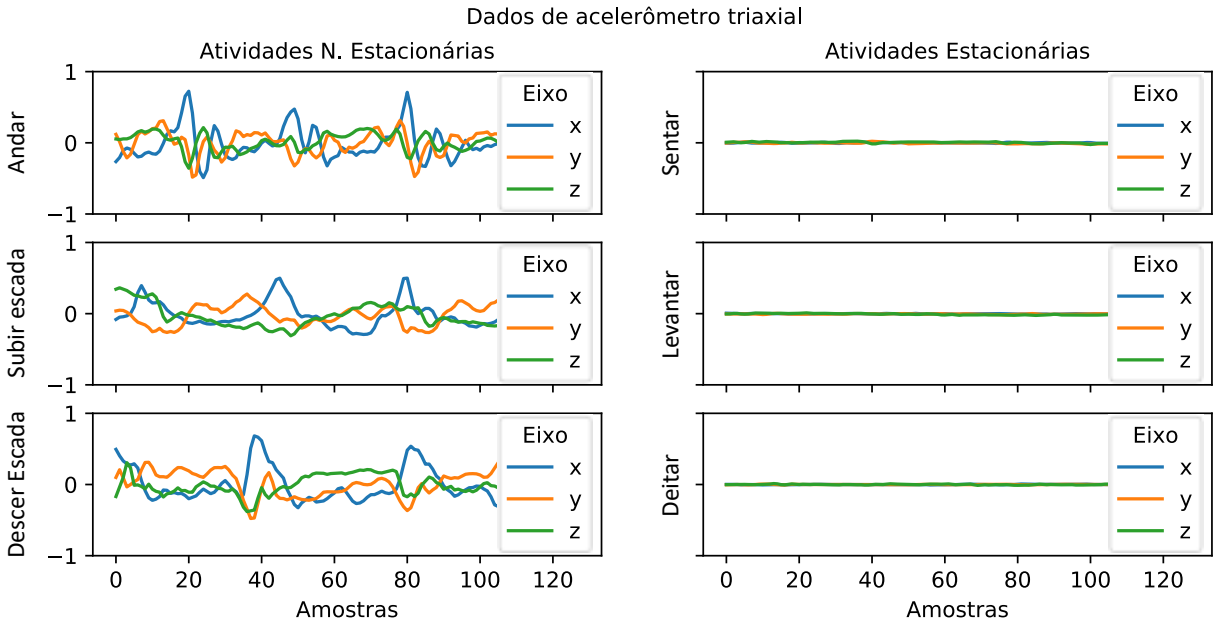


Figura 2 – Exemplo de dados de acelerômetro triaxial para 6 atividades distintas da base UCI-HAR dividias em atividades não estacionárias e atividades estacionárias.

É importante observar que os exemplos de dados da figura acima possuem duas subcategorias, atividades não estacionárias e atividades estacionárias. O efeito da similaridade interclasse pode ser observado sobre entre as classes dessas duas subcategorias, pois atividades estacionárias e não estacionárias possuem distribuição de dados semelhantes quando observadas como subgrupos.

Outro fator importante sobre os dados de sensores é a taxa de amostragem, a qual representa o número de capturas realizadas pelo sensor em um ciclo de tempo. Taxas de amostragem pequenas podem acarretar baixa acurácia nos modelos de aprendizagem de máquina por serem insuficientes para a generalização do modelo treinado, por outro lado, taxas de amostragem altas demais aumentam a complexidade dos dados e o custo para sua geração e armazenamento (SHOAIB *et al.*, 2015). A utilização de taxas a partir de 20Hz até 100Hz é comum na literatura e, apesar de que 20Hz pareça ser uma taxa singela, movimentos como andar, cair, levantar e sentar podem ser representados com taxas abaixo de 20Hz dado um sensor de acelerômetro (Karantonis *et al.*, 2006).

Portanto, levando em consideração o contexto da aplicação, os sensores a serem utilizados devem ser definidos, assim como a taxa de amostragem apropriada. A partir de então, os dados são coletados e apresentados em sua forma bruta. Como consequência, é necessário que esses dados sejam tratados para a próxima etapa no processo de RAH.

2.3 Pré-Processamento e Segmentação

De forma geral, os dados resultantes da aquisição, apresentados de forma bruta, precisam ser pré-processados visando a remoção de eventual ruído, adaptação e conversão para um formato homogêneo para que sejam utilizados nas etapas restantes. No pré-processamento, os dados resultantes normalmente estão dispostos através de *timestamps* associados às medições realizadas através dos vários sensores embarcados (KWAPISZ; WEISS; MOORE, 2011). Nessa fase, é importante trabalhar com remoção de valores nulos e identificação de ruídos não esperados, como remoção da interferência da aceleração gravitacional sobre sinais do acelerômetro triaxial (ANGUITA *et al.*, 2013).

Na sequência do processo, os dados dispostos em vários pontos associados apenas ao tempo (e.g. em uma tarefa com duração de 5 segundos capturada por um acelerômetro triaxial com taxa de amostragem de 100Hz, teríamos 500 instâncias com 3 valores reais associadas apenas a essa atividade) de forma contínua e intermitente precisam ser segmentadas para alimentar um modelo de aprendizagem de máquina.

2.4 Segmentação dos Sinais

Voltando para definição do problema de RAH simplificado apresentado na seção 2.1, quando buscamos reconhecer uma atividade, temos como resultado um conjunto $W = W_0, ..., W_{m-1}$ de tamanho fixo m. Nesse contexto, cada W_i representa um segmento da série temporal que contém os dados de todos sensores utilizados (e.g. o uso do acelerômetro triaxial indica que cada janela contém 3 séries temporais) de tamanho m, que por sua vez representa o número de amostras capturadas. Por exemplo, dada uma janela de 5 segundos com amostragem de 20Hz, teríamos uma janela de tamanho 100.

O tamanho da janela (m) a ser utilizada deve ser grande o suficiente para a classificação correta da atividade e pequena o bastante para não aumentar o tempo de detecção e recursos gastos. Dessa forma, a escolha adequada é dependente da atividade monitorada, pois, atividades simples, como andar e correr, são executadas em um curto espaço de tempo, necessitando assim de uma janela de poucos segundos (SHOAIB *et al.*, 2014). Por outro lado, atividades como preparar café ou fazer um sanduíche podem ser consideradas atividades complexas, até mesmo compostas por sub-atividades simples, e demandam mais tempo para o reconhecimento (BANOS *et al.*, 2014).

Uma vez fixado o tamanho da janela e conhecida a taxa de amostragem dos sensores, é possível realizar a segmentação dos dados. Neste processo, duas abordagens são mais comuns, o uso de janela simples e o uso de janela deslizante, ambas ilustradas na Figura 3. Na segmentação por janela simples, a série temporal é dividida em segmentos fixos, sem sobreposição de dados.

Apesar de ser uma abordagem rápida por sua simplicidade, no caso em que uma

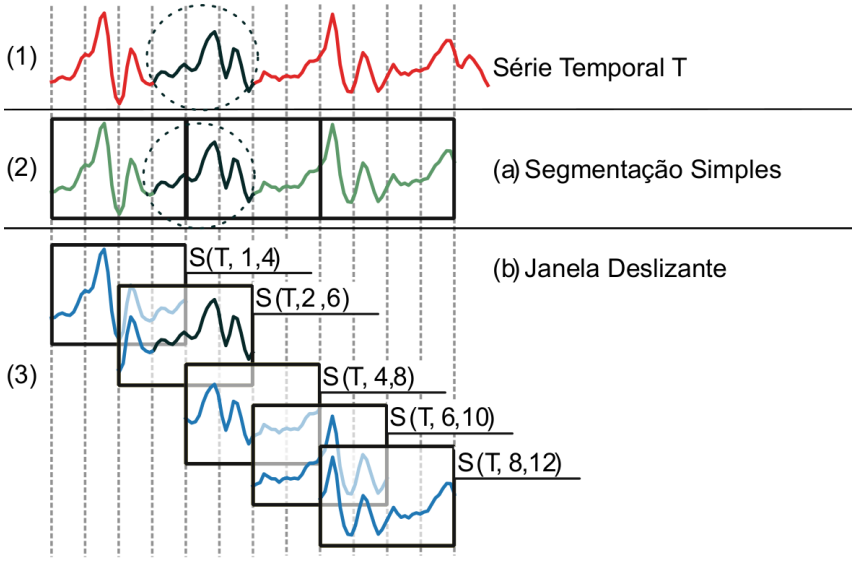


Figura 3 – (a) segmentação simples de uma série temporal. (b) segmentação com janela deslizante com 50% de sobreposição. Adaptado de BraganÇa (2019).

atividade esteja dividida entre duas janelas, ilustrada na Figura 3 na linha (1) (região em preto da série e destacada por um círculo pontilhado), o uso da janela sem sobreposição pode representar perda de informações (atividade dividida entre duas janelas em (2)). A técnica de janela deslizante com sobreposição por outro lado, é capaz de capturar melhor o segmento circulado (segmento S(T,2,6) em (3)), fornecendo um método robusto para segmentação de séries temporais (BANOS *et al.*, 2014).

Portanto, no processo de segmentação de dados de sensores em *smartphones* é necessário a escolha adequada do tamanho da janela e técnica de segmentação adotada. Uma vez segmentados, os dados passam para fase de extração de características.

2.5 Redes Neurais Profundas

Dentre os métodos de aprendizagem profunda aplicada no RAH, serão abordados os mais utilizados nos últimos anos na área. É importante salientar, que existem diversas redes neurais híbridas que combinam partes de diversas arquiteturas e, por não terem sido exploradas nesta pesquisa até o momento, não serão abordadas.

2.5.1 Rede Neural Convolucional

As RNCs são amplamente utilizadas em processamento de linguagem natural e Visão Computacional (NAKANO; CHAKRABORTY, 2017). São inspiradas nos neurônios encontrados no cérebro do gato. As RNCs possuem vantagem de compartilhamento de parâmetros, interações esparsas e representação equivalente. Uma vez que as camadas não são totalmente conectadas, o número de parâmetros é reduzido em relação a uma rede totalmente conectada e, dessa forma a rede se torna mais rápida e pode ser treinada mais

facilmente. Além disso, as operações de kernel funcionam como filtros locais e capturam a correlação espacial local existente nos dados (POUYANFAR *et al.*, 2018).

Normalmente, uma RNC é composta por camadas de convolução seguidas de camadas de *pooling* (sub-amostragem), ao fim, camadas totalmente conectadas seguidas de uma camada *Softmax*, a qual infere a classe. As camadas convolutivas possuem diversos *kernels* que são responsáveis por capturar características da imagem (matriz de entrada, a qual pode ser um conjunto de sensores) e as camadas de *pooling* fazem uma redução da dimensão das amostras para diminuir o número de parâmetros da rede. Um exemplo de uma RNC pode ser observado na Figura 4.

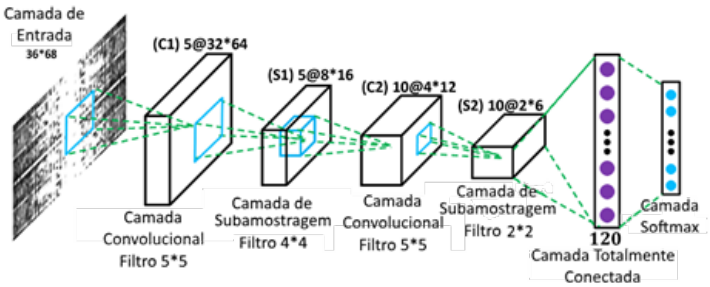


Figura 4 – Exemplo de Rede Neural Convolucional. Adaptado de Jiang e Yin (2015).

2.6 Otimização de hiperparâmetros: *Hyperband*

Os hiperparâmetros são entradas de um modelo de aprendizado de máquina que governam como que o modelo desempenha na generalização para predição de um novo dado, não visto no treinamento. Exemplos de hiperparâmetros são os números de filtros, tamanho do filtro de um RNC, a taxa de regularização utilizadas, a quantidade de unidades na camada oculta ou a taxa de aprendizagem utilizadas. A qualidade do modelo está diretamente relacionada ao conjunto de hiperparâmetros utilizados no treinamento, mas a relação entre estes é pouco conhecida quanto ao impacto no modelo resultante (Li *et al.*, 2016).

A otimização de hiperparâmetros é o processo em que conjuntos válidos de hiperparâmetros são avaliados com o objetivo de obter o modelo com melhor desempenho entre todas as configurações possíveis. Duas das técnicas utilizadas para essa otimização é o *grid search* e a busca aleatória (Yu; Zhu, 2020). A desvantagem geral quando levamos em consideração o *grid search* é que o custo computacional de treinar totalmente todas as configurações possíveis de hiperparâmetros se torna impraticável, pois o tamanho do espaço de busca cresce exponencialmente conforme cada possibilidade de hiperparâmetro avaliado, o que limita o tamanho de espaço de busca para se adequar aos recursos computacionais disponíveis. Já a busca aleatória possui a vantagem de percorrer o espaço de

busca usando distribuições sobre possíveis hiperparâmetros e a busca continua até que o recurso computacional seja escasso ou a métrica de avaliação desejada seja atingida.

Ambos os métodos apresentam desvantagens sobre o aspecto de gasto de recursos computacionais sobre conjuntos de configurações que não resultam em resultados esperados. Por exemplo, quando uma taxa de regularização muito alta presente em qualquer configuração de hiperparâmetros no espaço de busca, os dois métodos continuarão a gastar recursos computacionais em configurações onde está alta regularização está presente. Nesse cenário, o uso do *hyperband* é vantajoso, pois ele aloca recursos computacionais para subconjuntos de configurações com melhor desempenho em relação a função objetivo definida.

O *hyperband* utilizada a estratégia de divisão ao meio sucessiva aloca recursos computacionais para um subconjunto de configurações de hiperparâmetros obtidos por uma distribuição de amostragem uniforme com base no recurso computacional disponível, treina todos os modelos derivados deste subconjunto n^1 épocas, e a porção $1/n$ modelos com melhor desempenho de acordo com a função objetivo escolhida é mantida. Posteriormente, a melhor metade é treinada por n^2 épocas, garantindo que maior recurso computacional seja gasto em configurações com melhor desempenho. Novamente a porção $1/n$ deste subconjunto que melhor desempenhou em relação a função objetivo é mantida e o processo de divisão meio a meio e repetida k vezes, até que n^k = máximo de épocas.

2.7 Destilação de Conhecimento

As redes neurais profundas têm sido empregadas com sucesso em muitos domínios principalmente devido à sua escalabilidade para tratar dados em larga escala. No entanto, essas redes de alto desempenho são complexas, com milhares de parâmetros. Uma vez treinados, uma grande desvantagem de modelos tão amplos e profundos é que eles resultam em tempo de inferência elevado, pois precisam realizar um grande número de multiplicações. Além disso, ter uma grande quantidade de parâmetros torna os modelos mais exigentes em memória. Por esses motivos, redes complexas de alto desempenho podem não ser adequadas para aplicações com limitações computacionais, como memória, processamento, energia e latência (HINTON *et al.*, 2015; ROMERO *et al.*, 2014).

Os modelos profundos em larga escala alcançaram sucessos esmagadores, no entanto, a enorme complexidade computacional e os enormes requisitos de armazenamento tornam um grande desafio implantá-los em aplicativos em tempo real, especialmente em dispositivos com recursos limitados (GOU *et al.*, 2021).

Para mitigar essas limitações, uma nova forma de treinar redes neurais se baseia no conceito de destilação de conhecimento. A destilação de conhecimento é uma técnica para transferência de aprendizagem, onde o conhecimento de uma rede neural profunda

pré-treinada é destilado e transferido para outra arquitetura de rede neural, que pode ser dezenas ou até centenas de vezes mais simples (Hinton; Vinyals; Dean, 2015; YIM *et al.*, 2017).

Nesse novo modelo de treinamento, não é necessário se preocupar a priori com o modelo principal, chamado de professor. O modelo professor será responsável por desempenhar com excelência a tarefa para o qual foi treinado. Com esse objetivo, o modelo professor pode ser complexo, inclusive composto por *ensembles* de modelos. Uma vez que o modelo professor é treinado, podemos transferir o conhecimento adquirido pelo modelo professor para um modelo pequeno que é mais adequado para implantação (modelo estudante) em um processo conhecido como "destilação" (HINTON *et al.*, 2015).

A vantagem de escolher extrair conhecimento do modelo professor ao invés de treinar o modelo estudante independente (*from scratch*) é que esse modelo possui as estruturas e relações que são importantes para seu domínio. A literatura mostra que quando comparamos a rede estudante e a rede original e independente ((*from scratch*)) com a mesma configuração da rede estudante, mas treinada sem o auxílio da rede professor, a técnica de destilação de conhecimento apresenta três fenômenos importantes (HINTON *et al.*, 2015; ROMERO *et al.*, 2014; YIM *et al.*, 2017): (i) a rede estudante que aprende através da destilação tem melhor tempo de convergência se comparada ao modelo *from scratch* (não consideando o tempo do modelo professor ser treinado); (ii) a rede estudante supera a rede *from scratch*; e (iii) a rede estudante pode aprender o conhecimento destilado de uma rede professor treinada em uma tarefa diferente, e a rede estudante supera a rede original treinada do zero (YIM *et al.*, 2017).

Na destilação do conhecimento, um modelo estudante é geralmente supervisionado por um modelo professor. A ideia principal é que o modelo estudante mimetize o modelo professor para obter um desempenho competitivo ou superior. O problema-chave é como transferir o conhecimento de um modelo professor para um modelo estudante, que em geral tem arquitetura mais simples. Basicamente, uma arquitetura de destilação de conhecimento é composta por três componentes-chave: conhecimento, algoritmo de destilação e arquitetura professor-estudante. Uma estrutura geral professor-estudante para a destilação do conhecimento é mostrada na Figura 5.

Uma forma de transferir a capacidade de generalização do modelo professor para um modelo estudante é usar as probabilidades de classe produzidas pelo modelo professor como *soft labels* ou "alvos suavizados" para treinar o modelo estudante. Para este estágio de transferência, podemos usar o mesmo conjunto de treinamento ou um conjunto de "transferência" separado. Quando os *soft labels* têm alta entropia, eles fornecem muito mais informações por caso de treinamento do que os chamados *hard labels* e muito menos variação no gradiente entre os casos de treinamento, de modo que o modelo estudante pode ser treinado com muito menos dados do que o modelo original *from scratch* e usando

uma taxa de aprendizado muito mais alta. A Figura 5 mostra o processo de destilação de conhecimento de um modelo professor para um modelo estudante através de uma função de perda.

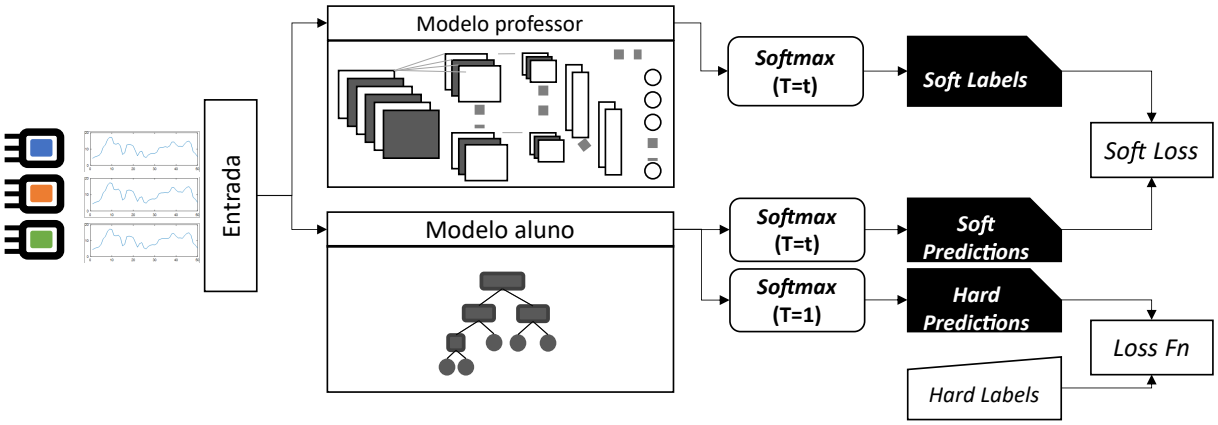


Figura 5 – Funcionamento do modelo de destilação de conhecimento de um modelo professor para um modelo estudante.

As redes de estudantes leves resultantes podem ser facilmente implantadas em aplicativos como reconhecimento visual, reconhecimento de fala e processamento de linguagem natural (NLP).

2.7.1 O processo de destilação de conhecimento

Nesta seção exploramos a definição de destilação de conhecimento proposta por Hinton *et al.* (2015), que treina uma rede estudante a partir da saída suavizada de uma rede professor, maior e mais complexa. A ideia é permitir que a rede estudante capture as informações fornecidas pelos rótulos verdadeiros e a estrutura mais refinada aprendida pela rede professor.

O processo pode ser resumido da seguinte forma (ROMERO *et al.*, 2014): seja uma rede de professores com uma saída *softmax* $P_t = softmax(a_T)$ onde a_T é o vetor de ativações pré-*softmax* do professor. No caso em que o modelo professor é composto por uma única rede, a_T representa as somas ponderadas da camada de saída. No entanto, se o modelo do professor é o resultado de um ensemble, ambos P_T e a_T são obtido pela média das saídas das diferentes redes.

O próximo passo é definir a rede estudante. Seja S uma rede estudante com parâmetros W_S e probabilidade de saída $P_S = softmax(a_S)$, onde S_i é a saída pré-*softmax* do estudante. A rede estudante será treinada de forma que sua saída P_S seja semelhante à saída P_T da rede professor, bem como aos rótulos y_{true}. Como P_T pode estar muito próximo da representação *hot code* do rótulo verdadeiro da amostra, uma relaxação $\tau > 1$ é introduzida para suavizar o sinal proveniente da saída da rede professor e, dessa

forma, fornecer mais informações durante o treinamento. A mesma relaxação é aplicada à saída da rede estudante P_S^τ, quando comparada à saída suavizada da rede professor P_T^τ, conforme apresentada na Equação 2.1.

$$P_T^\tau = softmax(\frac{a_T}{\tau}), P_S^\tau = softmax(\frac{a_S}{\tau}) \tag{2.1}$$

A rede do estudante é treinada para otimizar a função de perda apresentada na Equação 2.2.

$$L_{KD}(W_S) = \alpha * \mathcal{H}(y_{true}, P_S) + \beta * \mathcal{KLD}(P_T^\tau, P_S^\tau) \tag{2.2}$$

onde $\mathcal{H}$ se refere à entropia cruzada, KLD é a função de perda utilizada entre as predições do modelo professor e estudante suavizadas e, por fim, α e β são variáveis que controlam a participação da função de perda do modelo estudante e da função KLD. Observe que o primeiro termo da Equação 2.2 corresponde à entropia cruzada tradicional entre a saída de uma rede (estudante) e rótulos, enquanto o segundo termo utiliza uma função para comparar as predições suavizadas do modelo professor e modelo estudante. Podemos observar também que quando $\beta = 0$, temos uma função de perda final referente apenas à entropia cruzada, e quando $\alpha = 0$, apenas KLD contribui para função de perda final da destilação.

A função de perda utilizada na destilação de conhecimento tem como objetivo comparar quão próxima está a predição feita pelo modelo professor com a predição realizada pelo modelo estudante. Nesta pesquisa, utilizamos a divergência de Kullback-Leibler (KLD), a qual quantifica quanto uma distribuição de probabilidade difere de outra distribuição. Para um vetor de distribuição de probabilidade P_T^τ e um segundo vetor P_S^τ, podemos denotar a divergência entre os dois da seguinte forma:

$$KL(P_T^\tau || P_S^\tau) = \sum_{i=1}^{N} P_T^\tau(x_i) * log \frac{P_T^\tau(x_i)}{P_S^\tau(x_i)} \tag{2.3}$$

O resultado pode assumir valores entre 0 a ∞. Quando menor o valor resultante mais próxima a distribuição de probabilidade P_T^τ) é da distribuição P_S^τ).

3 TRABALHOS RELACIONADOS

Uma vez que a literatura pouco explora KD no contexto de *smartphones* (TANG *et al.*, 2021; CHEN *et al.*, 2018), adicionamos trabalhos que estão inseridos no reconhecimento de atividades humanas em tarefas de visão que contribuem para esta pesquisa (VU; LE; WANG, 2021; NI *et al.*, 2022). Além disso, apresentamos dois trabalhos de RAH, Ignatov (2018) e Peppas *et al.* (2020) que são usados como *baselines* para comparação dos resultados dos experimentos. A separação dos trabalhos relacionados em destilação de conhecimento no contexto de reconhecimento de atividades e o estado-da-arte no reconhecimento de atividades humanas buscas elucidar trabalhos que contribuíram para o desenvolvimento desta pesquisa.

3.1 Destilação de conhecimento no contexto de reconhecimento de atividades

O uso de algoritmos de ML rasos associados a características extraídas manualmente apropriadas de dados de sensores são capazes de atingir boa performance no RAH. Entretanto, o uso de RNP é capaz de extrair informações adicionais com os dados brutos (apenas segmentados) de sensores. Em sua pesquisa, Chen *et al.* (2018) empregam o uso de características estatísticas extraídas manualmente do domínio de tempo e de frequência dos dados de sensores de *smartphones* como entrada para uma rede neural de camada única como M_P. O modelo estudante M_E por sua vez, é composto por uma RNP com camadas de recorrência empilhadas para extrair características temporais existentes nos dados brutos dos sensores do *smartphone*. O processo de destilação é realizado no contexto de muti-visualização dos dados, enquanto o M_P recebe características extraídas manualmente do domínio de tempo e frequência, o M_E recebe dados brutos dos sensores do *smartphone*. Em teste realizado na base de dados UCI, o M_P atingiu uma acurácia de 96.5% e o M_E atingiu uma acurácia de 91.5% quando treinado sem o processo de destilação. Por fim, quando treinado com a destilação de conhecimento, o M_E destilado superou a acurácia do M_P ao atingir 97.7% de acurácia.

O processo de rotular dados capturados usando sensores de *smartphones* tem um custo associado pois é feito manualmente por um ser humano e, diferente de dados de áudio e imagem que podem ser rotulados posteriormente, dados de sensores são rotulados com captura de vídeo associada ou coletas definidas para guiar o ser humano, limitando a captura a cenários pouco variados e laboratoriais. Os algoritmos de RNP são dependentes do volume de dados rotulados disponíveis, de forma que quanto maior o volume de dados, melhor a capacidade de generalização dos modelos treinados. Nesse contexto, Tang *et al.* (2021) propuseram um método semi-supervisionado, onde uma base de dados rotulada B_R é dividida em conjuntos de treino B_R^{treino} e teste B_R^{teste}, o modelo professor M_P é

treinado de forma supervisionada com B_R^{treino} . Posteriormente, M_P é usado para rotular uma base de dados B_{R+NR}, composta por B_R^{treino} e uma base de dados não rotulada B_{NR}. As predições de M_P sobre B_{R+NR} são então filtradas para manter apenas K amostras para cada classe de atividade presente em B_R com confiança maiores que um limiar C, formando assim uma base de dados intermediária B_I com rótulos $Y_{preditos}$ por M_P. Em seguida, são aplicadas 8 funções de transformações de sinais sobre B_I, gerando um banco de dados aumentado B_A com rótulos muti-tarefa compostos por $Y_{preditos}$ e 8 rótulos binários associados a cada transformação aplicada. A B_A resultante é uma base de dados multi-tarefa auto-supervisionada. Em seguida, o modelo estudante M_E é pré-treinado para reconhecimento de 9 tarefas: a classificação multi-classe de $Y_{preditos}$ e uma classificação binária para cada transformação de sinal presente em B_A (treinamento auto-supervisionado). Por fim, camadas CNN que compõe o núcleo do M_E pré-treinado são congeladas e apenas a tarefa de classificação multi-classe é mantida para um treinamento fino com B_R^{treino}.

Os autores realizaram diversos testes usando o método por eles proposto, e obtiveram uma melhora na acurácia do M_E treinado com o método para todos os caso quando comparado com o M_E treinado apenas com o método totalmente supervisionado. Em especial, para as bases de dados avaliadas nesta pesquisa UCI e WISDM, usando a base de dados MobiAct (VAVOULAS *et al.*, 2016), como base não rotulada, o método atingiu um *F1-Score* médio de 91.35% e 90.81% respectivamente.

Vu, Le e Wang (2021) propuseram um método de auto-aprendizagem usando destilação de conhecimento no contexto de RAH para vídeos com técnicas de aumento de dados. No método proposto pelos autores, um modelo RNP estudante M_E é treinado com a destilação de conhecimento de um modelo "pseudo-professor", definido como o melhor M_E sobre a métrica de perda na validação das épocas anteriores do treinamento. O uso de aumento de dados e destilação de conhecimento foi capaz de gerar resultados estado-da-arte para o RAH no contexto de vídeo. De modo geral, o método demonstrou ser capaz de aumentar a capacidade de generalização do modelo de classificação final quando comparado com outros trabalho que usam modelos supervisionados, modelos auto-supervisionados e outros modelos de destilação de conhecimento, tanto em base de dados pequenas como em base de dados grandes.

O trabalho de Ni *et al.* (2022) introduz um metodologia de multi-modal para o RAH, combinando um modelo professor M_P treinado com dados de vídeo para destilação de conhecimento em um modelo estudante M_E que recebe apenas dados de acelerômetro para a tarefa de classificação de atividades. Os autores estendem a função de perda da destilação proposta por Hinton, Vinyals e Dean (2015) com a adição da importância da relação estrutural e da informação semântica para lidar com a diferença modal entre os domínios de visão e sensores. Para as base dados avaliadas, o método proposto pelos

autores foi capaz de melhorar a acurácia no teste usando apenas os dados do acelerômetro. Quando comparado com o M_E treinado apenas com os dados de sensores, a destilação de conhecimento foi capaz de aumentar a acurácia em 2.1%, demostrando que o uso da destilação de conhecimento multi-modal entre visão e sensores pode representar um ganho significativo de acurácia.

3.2 Estado da arte em reconhecimento de atividades humanas

Ignatov (2018) utiliza um RNP com camadas de convolução para extração automática de características de sinais de sensores de *smartphone*. Posteriormente a camada de extração dos sinais brutos dos sensores, 561 medidas estatísticas extraídas com funções do domínio de tempo e frequência são concatenadas no modelo de RNP proposto para o processo de classificação. O uso de CNN com características estatísticas apresentam ganho significativo na acurácia quando comparado com a CNN sem essas características, pois a metodologia proposta é capaz de utilizar características extraídas pela CNN para encontrar padrões locais do sinal, enquanto que as medidas HC mantém informações globais sobre a forma do sinal perdidas durante a convolução. Como resultado, o método atingiu una acurácia de 93.32% para a base WISDM e 97.63% para a base UCI.

Usando a mesma metodologia, Peppas *et al.* (2020) utilizaram uma CNN com medidas estatísticas para o RAH com dados de sensores de *smartphones*. A principal diferença deste trabalho quando comparado com o trabalho de (IGNATOV, 2018) é que as camadas de convolução são mais profundas e foram selecionadas apenas 40 medidas HC, o que diminui 7 vezes o número de parâmetros total da RNP resultante. Apesar da redução do número de parâmetros, o método atingiu uma acurácia de 94.18% para a base de dados WISDM.

3.3 Discussão

Na comunidade de pesquisa em RAH, existem soluções robustas para o problema de reconhecimento de atividades humanas, como os trabalhos de Ignatov (2018) e Peppas *et al.* (2020) apresentados na Seção 3.2. Embora esses trabalhos forneçam resultados promissores, os modelos apresentados não são viáveis para implantação em dispositivos com limitações de recursos computacionais. Nossa proposta apresenta uma solução viável e precisa quando comparada com as soluções mais robustas como as apresentadas na Seção 3.2 e também de modo geral com os trabalhos encontrados na literatura de RAH. Ao aplicar técnicas de destilação de conhecimento, apresentamos uma abordagem que permite o desenvolvimento de modelos com excelente custo-benefício em relação ao consumo de recursos computacionais e prontos para o mundo real. Apresentamos modelos compactos e com desempenho similar os modelos apresentados nesta seção, viáveis para execução diretamente nos dispositivos vestíveis.

4 KD-HAR: DESTILAÇÃO DE CONHECIMENTO EM MODELOS DE RECONHECIMENTO DE ATIVIDADES HUMANAS

Este capítulo descreve o método proposto KD-HAR (*Knowledge Distillation for Human Activity Recognition*) para a compressão de RNPs baseado na técnica de destilação de conhecimento aplicada à modelos de reconhecimento de atividades humanas usando dados de sensores inerciais. Inicialmente será descrita uma visão geral do método proposto e, na sequência, cada um dos seus componentes será detalhado.

4.1 Visão Geral do método KD-HAR

O método proposto utiliza conceitos de aprendizagem profunda sobre o domínio de séries temporais para a compressão de modelos de reconhecimento de atividades humanas no contexto de dispositivos móveis. Os conhecimentos adquiridos por modelos professores, obtidos por meio de técnicas de otimização de hiperparâmetros, são transferidos para modelos de RNP estudantes com menor complexidade. O procedimento envolve a avaliação da acurácia e do número de parâmetros dos modelos estudantes. A finalidade da avaliação é obter um conjunto de modelos estudantes potenciais com base em uma métrica previamente escolhida. A Figura 6 fornece uma visão geral do método proposto.

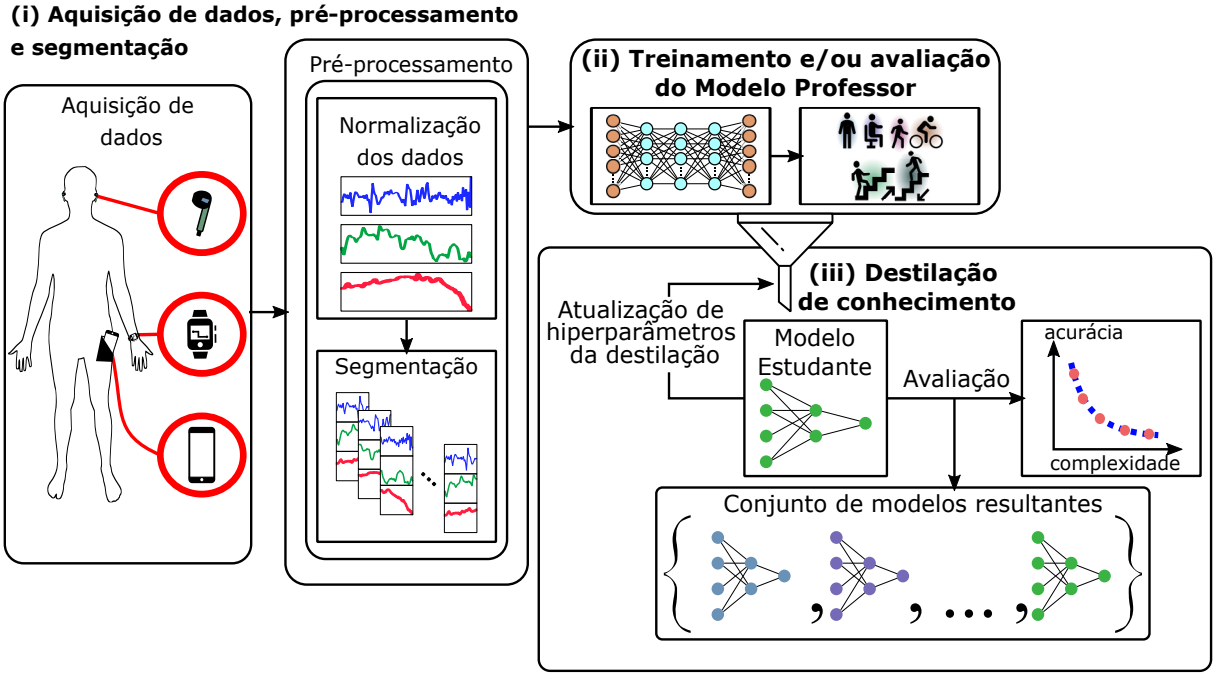


Figura 6 – Visão geral do método KD-HAR proposto para compressão de RNPs baseado na técnica de destilação de conhecimento aplicada à modelos de reconhecimento de atividades humanas.

O método KD-HAR é compostos das seguintes etapas: i) Aquisição e pré-processamento

dos dados; ii) Treinamento e/ou avaliação do modelo professor; iii) Processo de destilação do conhecimento do modelo professor para o modelo estudante.

Como exibido na Figura 6, o processo de RAH proposto inicia pela etapa de aquisição e pré-processamento dos dados, que tem como propósito adequar os dados brutos em um formato homogêneo para que possam ser utilizados na geração dos modelos de redes profundas. Na etapa seguinte são realizados o treinamento/escolha do modelo professor e a classificação das atividades para destilação do conhecimento para um modelo estudante, o qual é avaliado pela acurácia da classificação das amostras do teste e pelo o número de parâmetros do modelo. Após a avaliação, o modelo estudante é guardado em um conjunto de modelos resultantes e os hiperparâmetros da destilação são atualizados para um novo processo de destilação de conhecimento, buscando um conjunto final composto por modelos estudantes com maior acurácia. Detalhes de cada uma dessas etapas serão fornecidos nas seções seguintes.

4.2 Aquisição e Pré-processamento dos Dados

Como mencionado na Seção 2.2, a coleta de dados no domínio RAH é comumente realizada por meio de sensores inerciais presentes nos dispositivos vestíveis (e.g. *smartphones*, *smartwatches* e fones intra-auriculares inteligentes). Esses dados são capturados e armazenados em um formato bruto e, portanto, precisam ser pré-processados convertidos e adaptados em um formato homogêneo para que possam ser utilizados na geração dos modelos de redes profundas.

Neste trabalho os dados são tratado como séries temporais, visto que os sensores coletam dados ordenados no tempo. Isso permite que os sinais dos sensores possam ser segmentados em janelas de tempo. O processo de segmentação utilizado segue as recomendações encontradas em Banos *et al.* (2014) para problemas que envolvem o reconhecimento de atividades humanas. Nós adotamos a estratégia de janela deslizante com sobreposição de 50% para geração da base de dados usada no treinamento dos modelos. O tamanho da janela pode variar conforme o cenário avaliado. De forma geral, podemos definir que as atividades humanas avaliadas neste trabalho (por exemplo, andar, correr, deitar e parar) podem ser reconhecidas em um intervalo de tempo de 1 a 10 segundos.

4.3 Treinamento e Avaliação do Modelo Professor

O processo de destilação de conhecimento utilizado neste trabalho é baseado na abordagem introduzida por Hinton, Vinyals e Dean (2015), portanto, é necessário definir um modelo pré-treinado com o mesmo problema de classificação e com entrada derivável da base de dados estudada. Em outras palavras, o processo de destilação precisa de um modelo professor pré-treinado nas mesmas classes dos modelos estudantes alvo da

destilação. Assim, a base de dados tem que ter o mesmo formato para os dois conjuntos de modelos.

O modelo professor deve ser definido levando em consideração a melhor métrica de avaliação possível (por exemplo, acurácia ou *F1-Score*) sobre os dados de teste da base de dados escolhida. Além disso, é importante que seja garantido que o treinamento do modelo professor tenha sido feito usando a mesma divisão da base de treino e teste ou que a base de treinamento do modelo professor esteja contida na base de dados de treino que será utilizada para treinamento do modelo estudante.

Existem duas abordagens para escolher o modelo professor: fazer a escolha de um modelo pré-treinado (garantindo que não há viés em relação à base de dados de treino desse modelo) ou realizar o treinamento do modelo professor. Neste trabalho nós iremos treinar o modelo professor, buscando fornecer um bom modelo para o problema de reconhecimento de atividades humanas.

Para a escolha do modelo professor, este trabalho adota um método de otimização de hiperparâmetros, denominado de *hyperband*, que visa a seleção ótima de um conjunto de hiperparâmetros em um espaço de busca (LI *et al.*, 2018). A principal vantagem desse método é o tempo de execução quando comparado com outros métodos de otimização como *grid search* e *random search* tradicional. Ao invés de buscar conjuntos de parâmetros em todo espaço como faz o método *grid search* ou apenas aleatoriamente (e.g. *random search*) até satisfazer o resultado esperado, o *hyperband* utiliza um conjunto aleatório das combinações que são testadas seguindo heurística elitista para percorrer o maior número de combinações em um menor espaço de tempo.

Dada as especificações acima, o modelo professor proposto é composto por n blocos de convolução, de forma que cada bloco é composto por três camadas: uma camada de Convolução 1D, uma camada de *batch normalization*, uma camada de ativação $ReLU$; ao final uma camada de *Pooling 1D* (Médio ou Máximo) representando a saída do bloco, conforme mostrado na Figura 7. O objetivo principal do bloco é fazer a extração automática de características relevantes para o problema de classificação de atividades humanas.

Depois que os blocos de convolução são empilhados, a saída do último bloco é passada para uma camada de achatamento para preparar os dados para as demais camadas do modelo, pois a saída dos blocos de convolução possuem diversos canais (relacionado ao número de filtros da Camada de Convolução 1D). Essa camada de achatamento é seguida de uma estrutura de rede MLP, com uma camada Densa, uma camada de ativação $ReLU$, uma camada de *dropout* e, por fim, uma camada Densa com o número de neurônios referente a quantidade de classes do problema de classificação.

O conjunto de hiperparâmetros para cada camada, tal como o número n de blocos de convolução são selecionados com a otimização de hiperparâmetros *hyperband* e, para

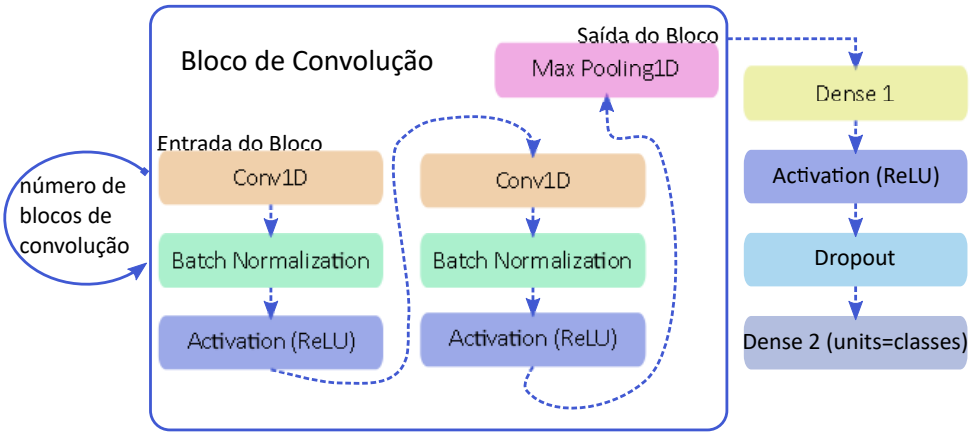


Figura 7 – A arquitetura usada para gerar o modelo professor.

cada camada eles podem variar conforme a Tabela 1.

Tabela 1 – Tabela do conjunto de hiperparâmetros e do número total de configurações dos hiperparâmetros que compõem o espaço de busca para o treinamento do modelo professor.

Tipo	**Hiperparâmetros**	**Conjunto**	Combinações
Bloco de Convolução	Quantidade de blocos	{1,2,3,4}	4
	Número de filtros	{32,64,...,224}	6
	Tipo de *Pooling*	{*Average*, *Maximum*}	2
	Número de unidades ocultas	{30,130,...,630}	6
Camada de *Dropout*	Fator de *dropout*	{0.05, 0.1,0.15,0.2}	4
	Tamanho do Espaço de busca		1152

Além dos hiperparâmetros que compõem o espaço de busca para o treinamento do modelo professor, outros parâmetros foram fixados observando o modelo proposto por Ignatov (2018) e são listados na Tabela 2. A fixação desses parâmetros foi necessária visto que a adição de mais hiperparâmetros no processo de otimização faz com que o tamanho do espaço de busca cresça exponencialmente, inviabilizando o processo de treinamento devido ao tempo e custo computacional.

O resultado da otimização de hiperparâmetros é um conjunto de modelos professores treinados. Os melhores resultados da otimização são então avaliados com a métrica escolhida para seleção do modelo professor final. Com o modelo professor candidato já treinado e avaliado, o próximo passo do método está relacionado com o processo de destilação de conhecimento para o modelo estudante.

4.4 Destilação de Conhecimento

A destilação de conhecimento é um procedimento para compressão de modelos de RNPs, no qual um modelo com menor tamanho (estudante) é treinado para corresponder a um modelo maior e robusto pré-treinado (professor). O conhecimento é transferido do

Tabela 2 – Hiperparâmetros fixos para otimização de hiperparâmetros e treinamento do modelo professor.

Tipo	**Hiperparâmetro**	**Valor**
Convolução 1D	Tamanho do *kernel* dos filtros	16
	Strides	1
	Padding	*same*
	Regularizador L2	0.0005
Pooling 1D	*Pool size*	4
	Strides	4
	Padding	*same*
Camada Densa 1	Regularizador L2	0.0005
Função de perda	*Categorical Crossentropy*	*from_logits = True*
Otimizador Adam	*Learning Rate*	0.001

modelo do professor para o estudante, minimizando uma função de perda, com o objetivo de combinar as predições de classe do professor (*logits/soft prediction*), bem como rótulos reais (*ground-truth/hard prediction*) do estudante (Hinton; Vinyals; Dean, 2015). Portanto, é necessário definir a arquitetura do modelo estudante de forma que o seu número total de parâmetros seja inferior ao número total de parâmetros do modelo professor.

A Figura 8 apresenta a arquitetura RNP estudante proposta, a qual mantém similaridades com a arquitetura do modelo professor. A arquitetura do modelo estudante é composta por m blocos de convolução. Cada bloco é composto por camadas de convolução 1D, *batch normalization*, ativação (ReLU) e uma camada de *Max Pooling1D*, de modo que, no bloco n a camada de Max Pooling1D é omitida. Por fim, é aplicada uma camada de *Global Average Pooling 1D* e os resultados são passados para uma camada de *Dropout*, uma camada Densa com ativação (ReLU) e uma camada densa com ativação linear corresponde ao final da arquitetura.

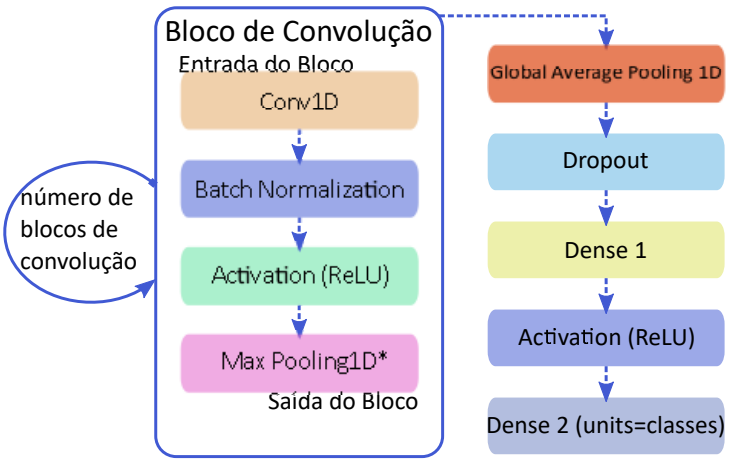


Figura 8 – A arquitetura de rede usada para gerar o modelo estudante.

A inserção de camadas de *Max Pooling* após cada camada de convolução reduz

o número de parâmetros quando comparamos com a estratégia que usa dois blocos de convolução da arquitetura do modelo professor. Além disso, também foi reduzido o número de blocos de convolução para 2 e o uso da camada de *Global Average Pooling* mantém apenas 1 valor por filtro usado na última camada de convolução, reduzindo consideravelmente o espaço de amostras quando comparamos ao processo de achatamento, que leva em consideração todas as amostras para todos os filtros da última camada de convolução. Por fim, a camada densa (*Dense 1*) é limitada a 1/4 do resultado da camada de *Global Average Pooling*.

Tabela 3 – Hiperparâmetros fixos para otimização de hiperparâmetros e treinamento do modelo aluno.

Tipo	**Hiperparâmetros**	**Conjunto**
Convolução 1D	Tamanho do *kernel* dos filtros	16
	Strides	1
	Padding	*same*
	Regularizador L2	0.001
Pooling 1D	*Pool size*	2
	Padding	*same*
Camada Densa 1	Regularizador L2	0.001
Camada *Dropout*	Fator de *dropout*	0.05
Função de perda	*Categorical Crossentropy*	*from_logits = True*
Otimizador Adam	*Learning Rate*	0.001

No modelo estudante, os parâmetros fixados diferem dos escolhidos no modelo professor somente no *pool size* para camadas de *pooling* e para o número de filtros da camada de convolução que são 32 filtros para primeira camada e 64 para a segunda. A Tabela 3 lista os hiperparâmetros fixados na geração do modelo estudante.

Assim que a arquitetura do modelo estudante é definida, o próximo passo para a destilação de conhecimento é definir os hiperparâmetros α e T (temperatura). Para definir esses parâmetros levamos em consideração a suavização das predições que ocorrem quando há a divisão do *logits* preditos pelos modelos antes da aplicação da função de *softmax*. Neste trabalho adotamos $\beta = 1 - \alpha$, sendo assim, o fator α está diretamente relacionado a importância que é dada a função de perda da destilação e a função de perda do modelo estudante. O valor de α aumenta a importância da participação da função de perda do modelo estudante de forma proporcional e diminui a participação da perda da destilação de forma inversamente proporcional.

Desta forma, um α pequeno representa menor contribuição da função de perda do estudante e maior contribuição para a função de perda da destilação. Em casos extremos, se $\alpha = 0$, apenas a função de perda da destilação é considerada. Quando $\alpha = 0.5$ obtemos um equilíbrio, onde ambas funções de perdas possuem a mesma importância. Por fim, quando α é 1, apenas a função de perda do modelo estudante é considerada, reduzindo o

processo para um treinamento sem destilação. A Figura 9 ilustra o processo de destilação.

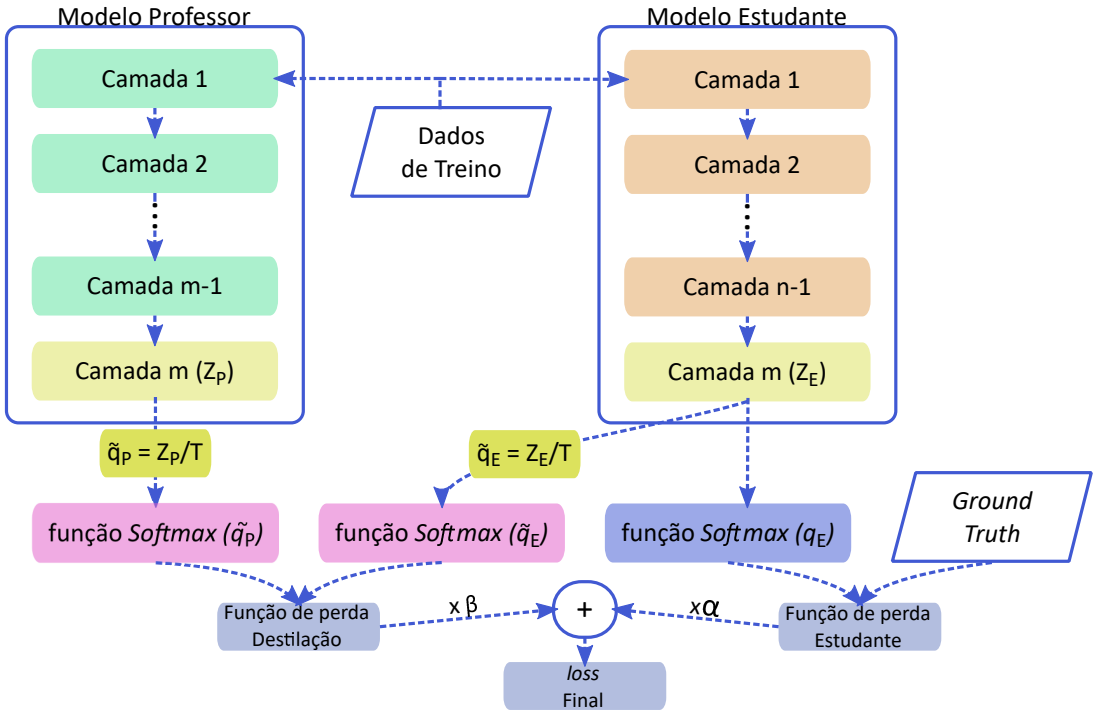


Figura 9 – Processo de destilação de conhecimento. A função de perda final é obtida levando em consideração uma função de perda sobre as probabilidades de classe dos professores e alunos suavizadas pela temperatura T, a qual é multiplicada por um fator β e somada a função de perda do aluno sem suavização multiplicada por α.

Considere um modelo professor de m camadas e um modelo estudante de n camadas, as camadas *m-1* e *n-1* contêm os vetores de *logits* Z_P e Z_E, dessa forma, é aplicada uma divisão pelo valor do parâmetro T (temperatura) para cada valor dos vetores Z. Posteriormente, uma função de *softmax* é aplicada sobre o vetor suavizado por T para obtenção de probabilidades. Para exemplificar o efeito da temperatura sobre o vetor Z, considere a predição de uma amostra da atividade *Andar* (Tabela 4) para a base de dados WISDM, onde *T=valor* é o valor referente a temperatura aplicada sobre o vetor Z, e *softmax(T=valor)* é o resultado da função *softmax* sobre o vetor suavizado.

Podemos observar que os *logits* possuem valores positivos e negativos, e sua soma é não é 1, diferente da função *softmax* aplicada sobre o vetor, onde a soma dos vetores é 1 e temos a probabilidade da amostra pertencer a todas as classes avaliadas. Quando há a suavização dos *logits* com o uso da temperatura e a função *softmax*, podemos observar que o vetor de probabilidade carrega mais informações entre as classes conforme a temperatura aumenta. Na Figura 10 é possível observar que quanto maior o valor da temperatura, menor é a probabilidade da amostra pertencer a classe *Andar*. Por outro lado, é possível visualizar que quando a temperatura é suficientemente alta, mais informações sobre similaridade interclasse é obtida. No exemplo, quando *T=10*, é possível observar que a atividade *Andar*

Tabela 4 – Vetores de *logits* suavizadas por T e vetores de probabilidade obtidos aplicando a função *softmax* sobre o vetor suavizado (*T=valor*) para uma amostra da classe andar predita por um modelo de RNP. Os valores dos *logits* suavizados foram arredondados para 2 casas decimentais e os valores do *softmax* para 3 casas decimais.

	Correr	**Andar**	**Subir**	**Descer**	**Sentar**	**Levantar**	**Soma**
t=1	-7.31	10.44	-3.61	-2.11	-10.39	-15.16	-28.14
softmax(t=1)	0.000	1.000	0.000	0.000	0.000	0.000	1.00
t=2	-3.65	5.22	-1.81	-1.05	-5.19	-7.58	-14.07
softmax(t=2)	0.000	0.997	0.001	0.002	0.000	0.000	1.00
t=5	-1.46	2.09	-0.72	-0.42	-2.08	-3.03	-5.63
softmax(t=5)	0.024	0.839	0.050	0.068	0.013	0.005	1.00
t=10	-0.73	1.04	-0.36	-0.21	-1.04	-1.52	-2.81
softmax(t=10)	0.089	0.526	0.129	0.150	0.065	0.041	1.00

predita pelo modelo é mais próxima de *Descer*, *Subir* e *Correr*. Enquanto isso, o vetor de probabilidade indica maior similaridade com as classes *Sentar* e *Levantar*. Por outro lado, quanto a temperatura é menor *(T=5)*, as classes *Sentar* e *Levantar* mantém uma relação de probabilidade em relação a classe real menor do que quando a temperatura é alta, enquanto isso, informações sobre similaridade com as classes *Descer*, *Subir* e *Correr* são mantidas.

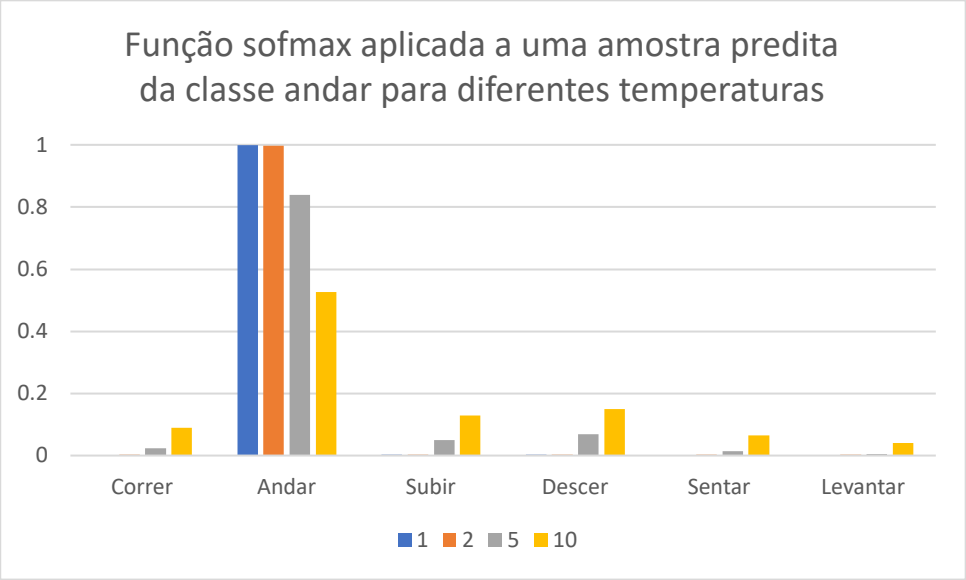


Figura 10 – Probabilidade de classes obtidas ao aplicar a função de *softmax* em um vetor de *logits* predito por um modelo de RNP para uma amostra da classe *Andar* da base de dados WISDM para as temperaturas 1, 2, 5, e 10.

No método de destilação proposto nós avaliamos o impacto da temperatura no processo de destilação de conhecimento e do hiperparâmetro α com as sugestões propostas por Hinton, Vinyals e Dean (2015), onde $\alpha = \{0.1, 0.2, 0.5\}$ e temperatura $(T) = \{1, 2, 4, 10\}$. A variação da temperatura busca encontrar um ponto onde as informações adicionais sobre a similaridade interclasse são suficientemente boas sem trazer prejuízo para o aprendizado

do modelo. Por sua vez, a variação do hiperparâmetro α busca encontrar um ponto de equilíbrio onde a importância da destilação é benéfica ao modelo estudante.

4.5 Considerações finais

Este capítulo descreve a abordagem utilizada para a compressão de RNPs usando destilação de conhecimento na área de RAH. Nós descrevemos o método de treino e avaliação para geração de modelos professores e modelos estudantes. O especialista, através de modificações (incremento ou decremento) no número de blocos de convolução, número de neurônios ou filtros em cada camada para reduzir ou aumentar o número de parâmetros da RNP estudante pode buscar um modelo suficientemente pequeno para a utilização em contexto de restrição de *hardware*. Além disso, a abordagem permite avaliar cada estudante proposto em relação aos hiperparâmetros utilizados na destilação de conhecimento e fornece uma ferramenta para decisão sobre a escolha de qual modelo estudante gerado utilizar de acordo com o problema de classificação avaliado.

5 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta os experimentos utilizados para avaliar o método KD-HAR na compressão dos modelos de reconhecimento de atividades humanas. O capítulo inicia descrevendo o protocolo experimental utilizado, incluindo a descrição das bases de dados públicas, o método de validação e as métricas para contagem de parâmetros e tamanho da rede neural profunda. Os resultados são divididos em três partes: (i) destilação de modelos na base de dados UCI; (ii) destilação de modelos na base de dados WISDM e; (iii) uma avaliação global dos resultados da abordagem sobre as duas bases de dados. Por fim, são realizadas algumas considerações finais sobre os resultados obtidos.

5.1 Protocolo Experimental

Esta seção apresenta uma visão geral de como os experimentos foram planejados, organizados e executados para analisar a capacidade de compressão do método proposto no contexto de HAR. Os experimentos foram agrupados em dois cenários:

- Avaliação do método proposto usando o *dataset* UCI-HAR. O objetivo é comparar o desempenho do método na compressão do modelo estudante proposto em relação ao modelo professor treinado.
- Avaliação do método o *dataset* WISDM. Neste cenário, além de avaliar o desempenho do método na compressão do modelo estudante, nós também comparamos os resultados dos trabalhos de (IGNATOV, 2018; PEPPAS *et al.*, 2020).

Para cada cenário, o modelo professor (KD-P) representa um modelo robusto de reconhecimento de atividades que é treinado e selecionado utilizando o método de otimização de hiperparâmetros *hyperband*. Logo em seguida, são treinados 12 modelos estudantes com a destilação de conhecimento (KD-S) e um modelo estudante sem a destilação de conhecimento (KD-0) (totalizando 13 modelos estudantes). O número de modelos estudantes treinados com a destilação de conhecimento é resultado da combinação dos hiperparâmetros α e temperatura T, como descrito na Seção 4.4. São avaliadas as temperaturas 1, 2, 5 e 10 e as variações de α de 0.1, 0.2 e 0.5, sendo assim, são treinados 12 modelos estudantes com a destilação de conhecimento.

5.1.1 Conjunto de dados

Os experimentos deste trabalho foram baseados em duas bases de dados comumente utilizadas na literatura para avaliar a classificação de atividades humanas: as bases de dados UCI-HAR de Anguita *et al.* (2013) e WISDM de Kwapisz, Weiss e Moore (2011).

A base de dados UCI-HAR foi construída a partir de dados coletados de 30 pessoas com idades que variam entre 19 e 48 anos. Cada pessoa executou 6 atividades físicas como andar, sentar, ficar em pé, deitar, subir e descer escadas. Os dados foram coletados de um celular Samsung Galaxy S2 utilizando os sensores acelerômetro e giroscópio a uma frequência de 50 Hz. A coleta foi realizada com o celular localizado na cintura dos usuários. A base de dados contém 10299 instâncias. Cada instância contém 561 atributos extraídos através de funções estatísticas. A base de dados é segmentada por meio da técnica de janela deslizante com sobreposição de 50% em um intervalo de 2.56 segundos (128 amostras por eixo de cada sensor).

A base de dados WISDM (*Wireless Sensor Dada Mining*) foi construída a partir de dados coletados de 36 pessoas. Cada pessoa, de posse de um smartphone, executou 6 atividades físicas como correr, andar, subir as escadas, descer as escadas, sentar e levantar. Os dados brutos são compostos por 1,098,207 instâncias, cada uma com 3 atributos. Nos experimentos executados, as bases de dados foram segmentadas com janela deslizante com 50% de sobreposição de dados. Não foram aplicados nenhum pré-processamento extra para as bases de dados.

A Tabela 5 apresenta um resumo das características das bases de dados utilizadas nos experimentos.

Tabela 5 – Sumário sobre as características das bases de dados e configurações da segmentação utilizadas nos experimentos.

Base de dados	**WISDM**	**UCI HAPT**
Nº de indivíduos	36	30
Frequência (Hz)	20	50
Tamanho da Janela	200 (10 s)	128 (2.56 s)
Tipo de Janela	Janela Deslizante	Janela Deslizante
	50% de sobreposição	50% de sobreposição
Instâncias	893702	748406
Instâncias Segmentadas	10393	10399
Distribuição das instâncias para cada classe	Levantar (4.3%)	Levantar (19%)
	Sentar (5.4%)	Sentar (17.3)
	Subir as escadas (10.1%)	Subir as escadas (14.8%)
	Descer as escadas (8%)	Descer as escadas (13.5%)
	Andar (40%)	Andar (16.5%)
	Correr (32%)	Deitar (14.8%)
Sensores	Acelerômetro 3D	Acelerômetro 3D, Giroscópio

5.1.2 Estratégia de validação

Este trabalho adota divisão da base e treino por indivíduos para avaliar a capacidade de generalização de um modelo de RAH, de forma que dados de um indivíduo que aparecem no treino não aparecem no teste. Neste trabalho nós adotamos a validação *hold-out* ao invés da validação cruzada. A validação cruzada com esta estratégia pode adotar partições k garantindo que indivíduos do treino não apareçam no teste, ou então, derivar uma

estratégia de validação cruzada por indivíduo, onde k partições equivale ao número de indivíduos na base de dados.

A desvantagem da validação cruzada para o método KD-HAR, tanto considerando k partições ou $k = indivíduos$, é o custo computacional associado ao treinamento de k modelos professores, e $k * 13$ modelos estudantes para cada conjunto de dados avaliado.

O uso do *hold-out* por indivíduo busca manter o conjunto de treino sem viés em relação ao conjunto de teste para avaliar a capacidade de generalização dos modelos treinados. A Tabela 6 apresenta os valores percentuais utilizados na divisão da base de treino e teste, tal como os usuários que foram inclusos em cada partição para as bases de dados utilizadas no experimento.

Tabela 6 – Valores de separação do *hold-out* utilizado e conjunto de usuários utilizados no treino e no teste para as bases de dados dos experimentos.

Base de dados	***Hold-Out***	**UCI**	**WISDM**
Conjunto de Teste	30%	[2, 4, 9, 10, 12, 13, 18, 20, 24] (9 usuários)	27,28...36 (10 usuários)
Conjunto de Treino	70%	Demais 21 usuários	Demais 26 usuários

5.1.3 Métricas de avaliação

Este trabalho usa a acurácia e a medida F sobre esses dados para analisar a abordagem proposta. O uso da acurácia é complementado pela medida F (*F-score* ou *F1-score*), a qual pode ser considerada uma métrica mais adequada para RAH por ser computada a partir de mais duas métricas, a precisão e a sensibilidade (Jordao *et al.*, 2018). Essas métricas são derivadas da matriz de confusão, uma matriz gerada com os valores reais e preditos pelo modelo. Considere a seguinte matriz de confusão na Tabela 7 para um problema de classificação de duas classes (Sim e Não):

Tabela 7 – Exemplo de matriz de confusão para duas classes (Sim e Não).

		Rótulo predito	
		Sim	*Não*
Rótulo verdadeiro	*Sim*	Verdadeiro Positivo	Falso Negativo
	Não	Falso Positivo	Verdadeiro Negativo

onde:

- Verdadeiro Positivo (VP): amostra real *Sim* predita como *Sim*;
- Falso Negativo (FN): amostra real *Sim* predita como *Não*;
- Falso Positivo (FP): amostra real *Não* predita como *Sim*;
- Verdadeiro Negativo (VP): Amostra real *Não* predita como *Não*.

A matriz de confusão é uma ferramenta importante pois permite avaliar o desempenho das predições do modelo de forma global. Além disso, é possível avaliar as predições do modelo e realizar com quais classes há confusão quando uma amostra é predita e perceber similaridades interclasses.

A acurácia representa o desempenho geral da classificação e pode ser calculada da seguinte forma:

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{5.1}$$

Podemos considerar analisar apenas os acertos da classificação, a métrica precisão considera apenas a amostras corretamente classificadas em relação ao número de amostras globais classificadas positivamente. Desta forma, é menos suscetível a representar resultados otimistas em predições onde há desbalanceamento de classes. A precisão pode ser obtida pela seguinte fórmula:

$$Precision = \frac{VP}{VP + FP} \tag{5.2}$$

A sensibilidade, por outro lado, considera as amostras preditas corretamente em relação ao número total de amostras da classe avaliada e pode ser obtida pela seguinte fórmula:

$$Sensibilidade = \frac{VP}{VP + FN} \tag{5.3}$$

Por fim, a *F1-score* combina a precisão e sensibilidade e avalia o desempenho geral do modelo sem manter a suscetibilidade a resultados otimistas quando há desbalanceamento de classe.

$$F1\text{-}Score = 2 * \frac{precição * sensibilidade}{precição + sensibilidade} \tag{5.4}$$

Além das métricas descritas acima, fazemos a comparação do número de parâmetros e do tamanho do modelo gerado. Para obter o número de parâmetros, são somados os parâmetros treináveis e não treináveis do modelo RNP gerado pelo *framework keras* (Figura 11). O tamanho final do modelo é obtido com o *Framework TensorFlow Lite*, o qual gera o modelo em formato adequado para exportação e utilização em dispositivos móveis. Como podemos observar na Tabela 8, o tamanho do modelo gerado para o formato lite difere do tamanho de um modelo treinado e salvo pelo *framework Keras*, por isso a conversão é necessária para avaliar o uso de espaço nos dispositivos móveis.

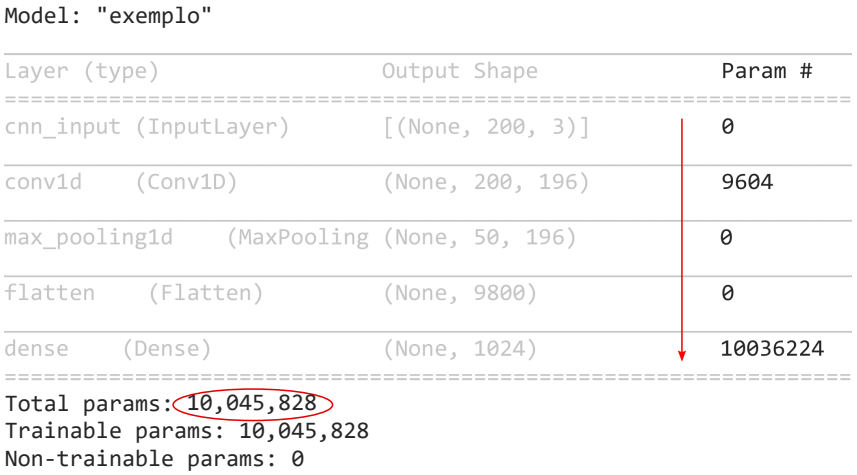


Figura 11 – Exemplo de número total de paramêtros retornado com o *Framework Keras*.

Tabela 8 – Espaço de armazenamento utilizado por um modelo RNP treinado por 100 épocas com 10 milhões de parâmetros quando salvo com o *framework Keras* e com o *framework Tensorflow Lite*.

	Modelo *Keras*	**Modelo *Tensorflow Lite***
Tamanho (MB)	114	38.3

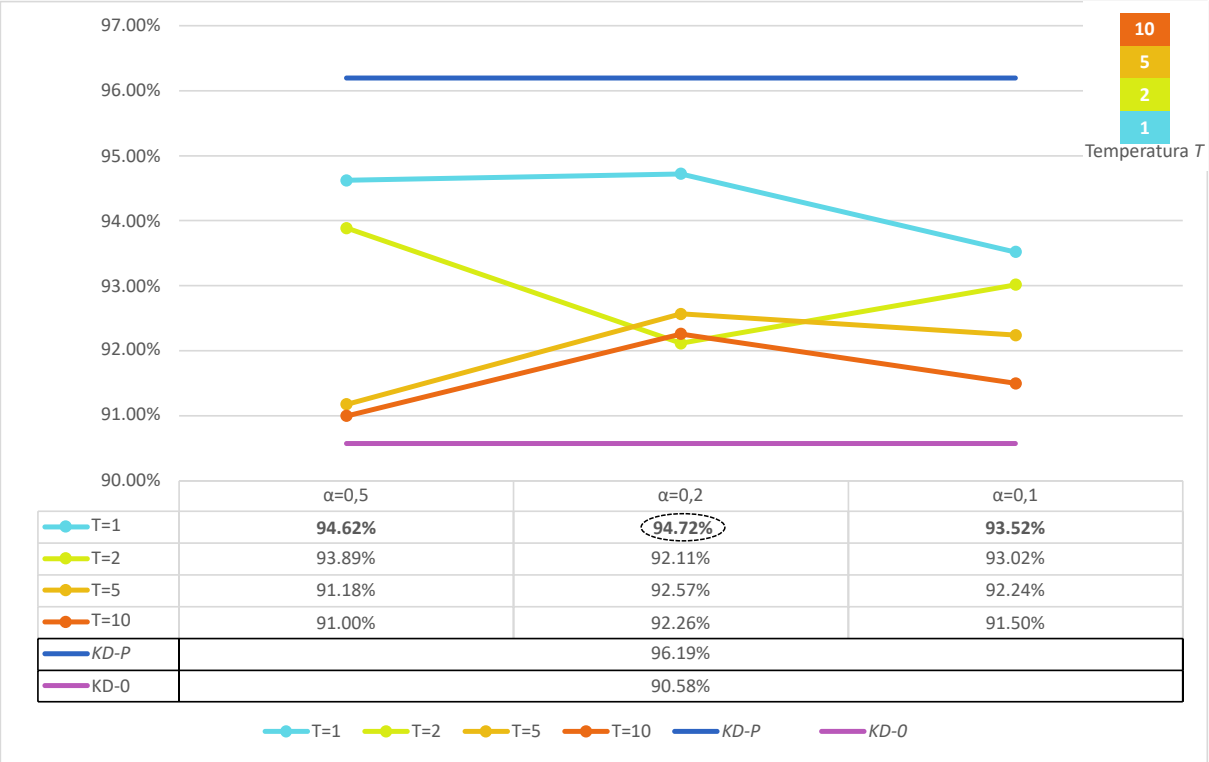
5.2 Resultados

Esta seção apresenta os resultados obtidos nesta pesquisa utilizando a destilação de conhecimento para compressão de modelos de classificação no contexto de reconhecimento de atividades humanas. Como mencionado anteriormente, os resultados são divididos em três partes: *(i)* destilação de modelos na base de dados UCI; *(ii)* destilação de modelos na base de dados WISDM e; *(iii)* uma avaliação global dos resultados da abordagem sobre as duas bases de dados.

5.2.1 Resultados para a base de dados UCI-HAR

A Figura 12 apresenta o desempenho do modelo *KD-P* (professor), em termos de acurácia, do modelo *KD-S* (estudante) e do modelo *KD-0* (*from scratch*) na base UCI. O modelo *KD-P* possuí 1.57 milhões de parâmetros enquanto o modelo *KD-S* possui 37.53 mil, cerca de 97% menor em número de parâmetros. Para todas as combinações de α e temperatura T avaliadas, o resultado da perda de desempenho na acurácia varia entre 1.47% a 5.16%, sendo menor que a perda de desempenho quando comparado com o modelo *KD-0* (5.61%). Isso mostra que o modelo *KD-S* mantém um bom desempenho, apesar da alta taxa de compressão de 42x em relação ao tamanho do modelo *KD-P*.

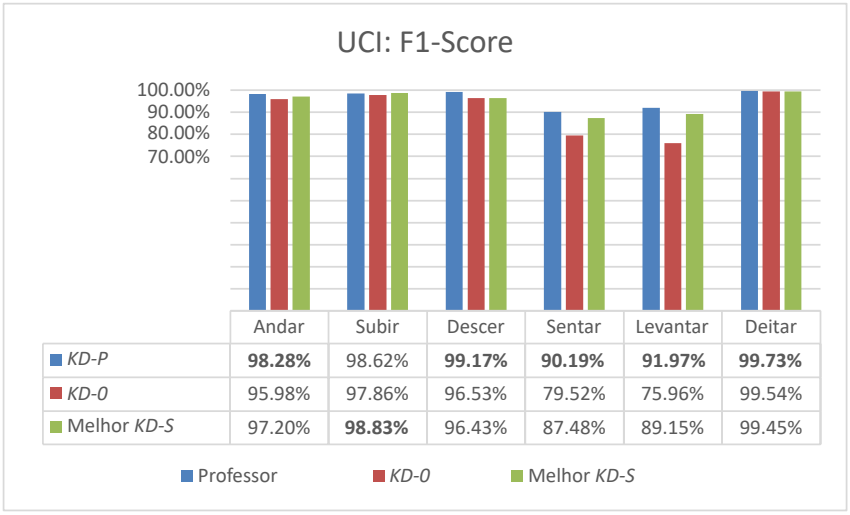
Conforme mostrado na Figura 12, a medida que a variável de temperatura T aumenta, a acurácia dos modelos estudantes tende a diminuir. As melhores acurácias são obtidas quando $T = 1$. Nos testes, a combinação de $\alpha = 0.2$ e $T = 1$ resulta no melhor



	α=0,5	α=0,2	α=0,1
T=1	**94.62%**	**94.72%**	**93.52%**
T=2	93.89%	92.11%	93.02%
T=5	91.18%	92.57%	92.24%
T=10	91.00%	92.26%	91.50%
KD-P	96.19%		
KD-0	90.58%		

Figura 12 – Acurácia do teste dos modelos estudantes destilados com variação da temperatura T e α, do modelo estudante treinado sem destilação e do modelo professor.

modelo estudante destilado (em relação a acurácia resultante da base de testes), o qual tem uma perda de apenas 1.47% de acurácia comparado ao modelo *KD-P* e um ganho de 4.14% de acurácia em relação ao modelo *KD-0*.



	Andar	Subir	Descer	Sentar	Levantar	Deitar
KD-P	**98.28%**	98.62%	**99.17%**	**90.19%**	**91.97%**	**99.73%**
KD-0	95.98%	97.86%	96.53%	79.52%	75.96%	99.54%
Melhor *KD-S*	97.20%	**98.83%**	96.43%	87.48%	89.15%	99.45%

Figura 13 – *F1-score* do teste do melhor modelo estudante destilado, do modelo estudante treinado sem destilação e do modelo professor.

A Figura 13 contém o resultado do melhor modelo *KD-S* (estudante destilado) em

comparação com o modelo *KD-P* e com o modelo *KD-0*, em termos de métrica *F1-Score*. O melhor modelo estudante destilado mantém o *F1-Score* muito próximo do modelo professor para as atividades não-estacionárias como andar, subir escadas e descer escadas. Por outro lado, observamos maior perda quando levamos em consideração as atividades estacionárias como sentar e levantar. De forma a complementar a análise, a Figura 14 apresenta as matrizes de confusão para os modelos *KD-P*, *KD-S* e *KD-0*.

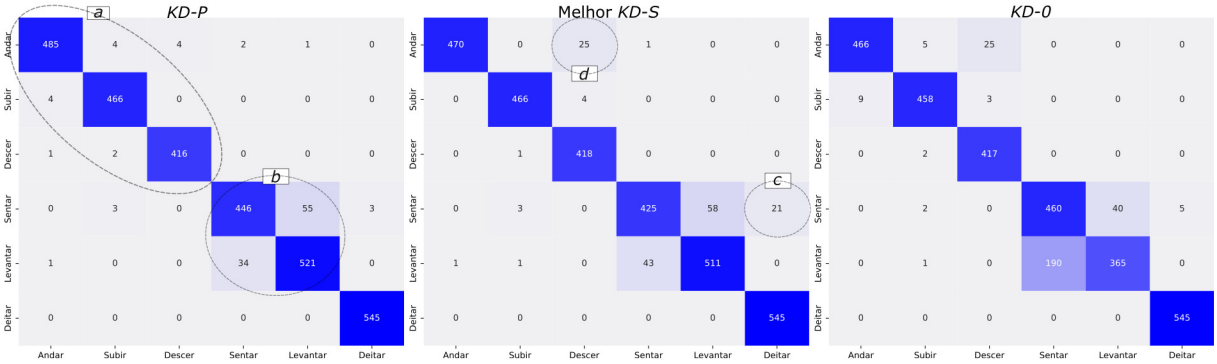


Figura 14 – Matriz de confusão do teste do modelo professor (*KD-P*), melhor modelo estudante destilado (*KD-S*) e do modelo estudante sem destilação (*KD-S*).

O modelo *KD-P*, *KD-S* e *KD-0* são capazes de diferenciar com facilidade as atividades não-estacionárias (andar, subir e descer as escadas; região *a* de cada matriz de confusão). No entanto, as atividades estacionárias apresentam um maior desafio, pois as atividades sentar e levantar são confundidas uma com a outra (região *b* das matrizes). A atividade deitar para o modelo *KD-S* passa a ser confundida com a atividade sentar, se comparado aos outros dois modelos (círculo *c*). Além disso, para os modelos estudantes *KD-S* e *KD-0*, houve a acentuação na confusão entre as classes andar e descer escadas (região *d* das duas matrizes).

5.2.2 Resultados para a base de dados WISDM

Diferentemente da base UCI-HAR, a base de dados WISDM possui maior desbalanceamento de classes e foi coletada fora de ambiente de laboratório. Por essas razões, ela apresenta um desafio maior para classificar atividades humanas quando comparada a base UCI-HAR. Os resultados obtidos nesta base, em termos de acurácia, para os modelos *KD-P*, *KD-S* e *KD-0* são apresentados na Figura 15.

Conforme apresentado na Figura 15, o modelo *KD-P* possuí 0.67 milhões de parâmetros e o modelo estudante (*KD-S*) 35.99 mil parâmetros. A taxa de compressão obtida foi de 18.7x em relação ao tamanho do modelo *KD-P*. Assim como no resultados apresentados para a base UCI-HAR, para todas as combinações de α e temperatura T, o desempenho dos modelos estudantes destilados superaram o modelo estudante treinado

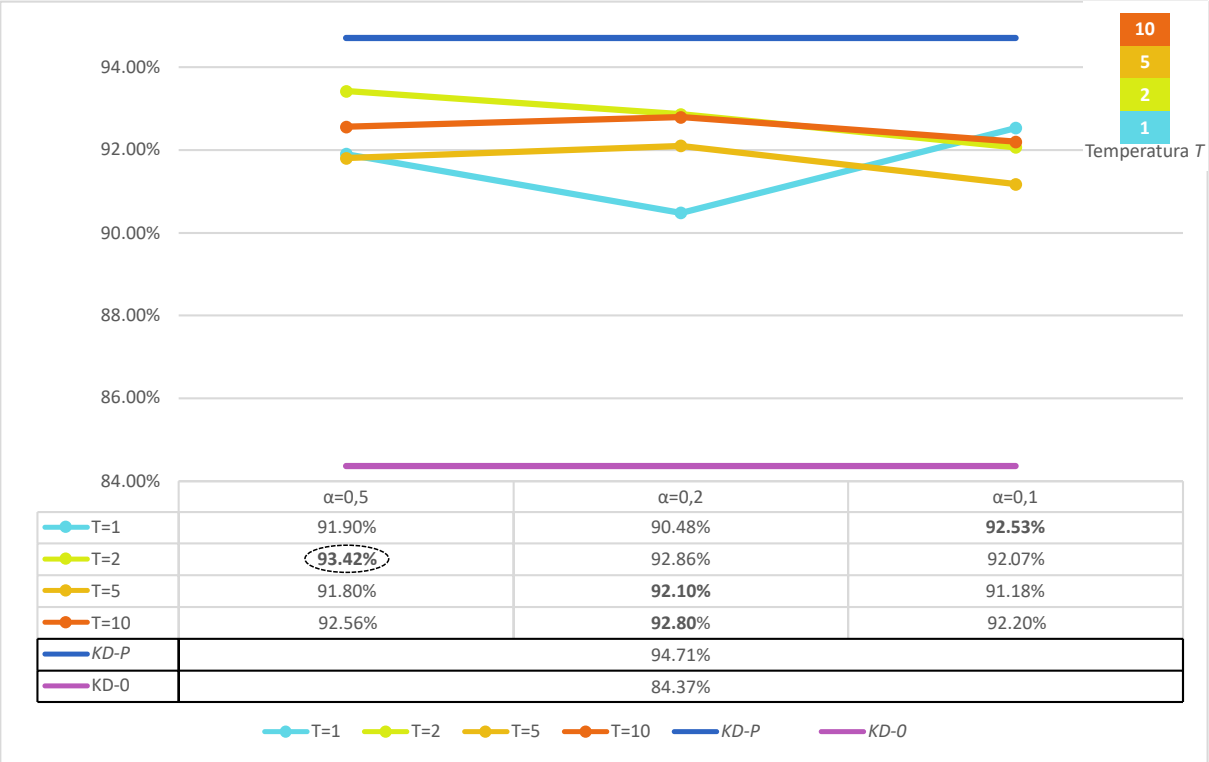


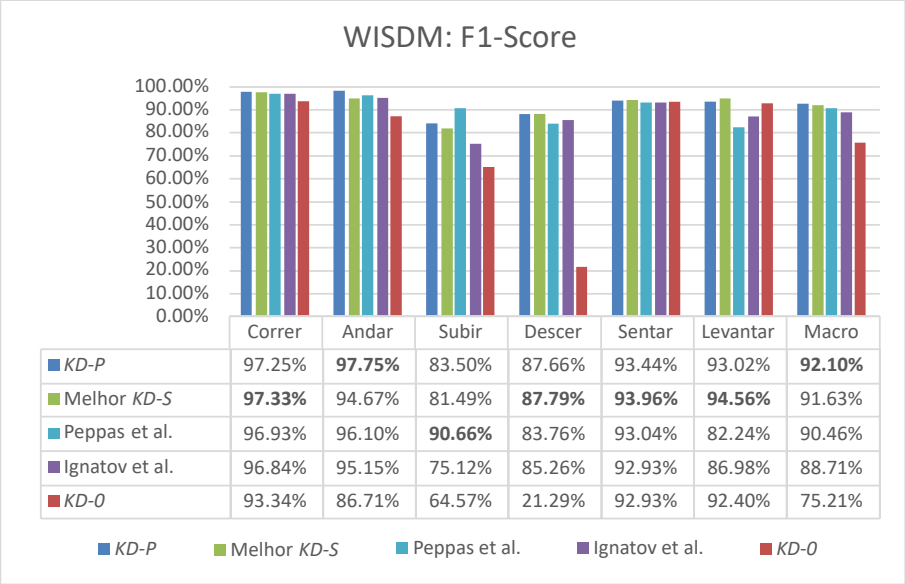
Figura 15 – Acurácia do teste dos modelos estudantes destilados com variação da temperatura e α, do modelo estudante treinado sem destilação (*KD-0*) e do modelo professor (*KD-P*).

sem destilação, mas desta vez, a margem de diferença foi maior, variando entre 6.11% à 9.05%.

Diferente dos resultados encontrados na base UCI, as probabilidades suavizadas por uma temperatura T diferente de *1* foram benéficas para a acurácia do modelo estudante destilado. A média de acurácia de $T = 2$ e $T = 10$ são próximas, mas o melhor modelo estudante destilado foi obtido com $T = 2$ quando $\alpha = 0.5$. Neste cenário, o modelo *KD-S* apresentou melhor resultado com maior participação da função de perda do aluno, a qual teve importância equiparada a função de perda da destilação.

A Figura 16 mostra o desempenho dos modelos, em termos de *F1-Score*, para o melhor modelo estudante destilado (*KD-S*), o modelo professor (*KD-P*) e o modelo estudante treinado sem destilação (*KD-0*). Também são introduzidos os resultados do trabalho de Ignatov (2018) e Peppas *et al.* (2020) para uma análise comparativa.

O modelo *KD-P* obtido pela otimização de hiperparâmetros apresenta maior macro *F1-Score* quando comparado com os demais modelos. Por sua vez, o modelo *KD-S* tem um *F1-Score* superior aos resultados dos trabalhos de Peppas *et al.* (2020) e Ignatov (2018) (1.17% e 2.92%) e também apresenta resultado superior ao modelo *KD-0* (16.42%), com ampla margem. As atividades subir e descer escadas apresentam as maiores perdas para



	Correr	Andar	Subir	Descer	Sentar	Levantar	Macro
KD-P	97.25%	**97.75%**	83.50%	87.66%	93.44%	93.02%	**92.10%**
Melhor *KD-S*	**97.33%**	94.67%	81.49%	**87.79%**	**93.96%**	**94.56%**	91.63%
Peppas et al.	96.93%	96.10%	**90.66%**	83.76%	93.04%	82.24%	90.46%
Ignatov et al.	96.84%	95.15%	75.12%	85.26%	92.93%	86.98%	88.71%
KD-0	93.34%	86.71%	64.57%	21.29%	92.93%	92.40%	75.21%

Figura 16 – *F1-score* do teste do melhor modelo estudante destilado (*KD-S*), do modelo estudante treinado sem destilação (*KD-0*), do modelo professor (*KD-P*) e dos trabalhos de Peppas *et al.* (2020) e Ignatov (2018).

todos os modelos. De forma a complementar, a Figura 17 apresenta as matrizes de confusão do modelo professor, do melhor modelo estudante destilado e do modelo estudante treinado sem destilação.

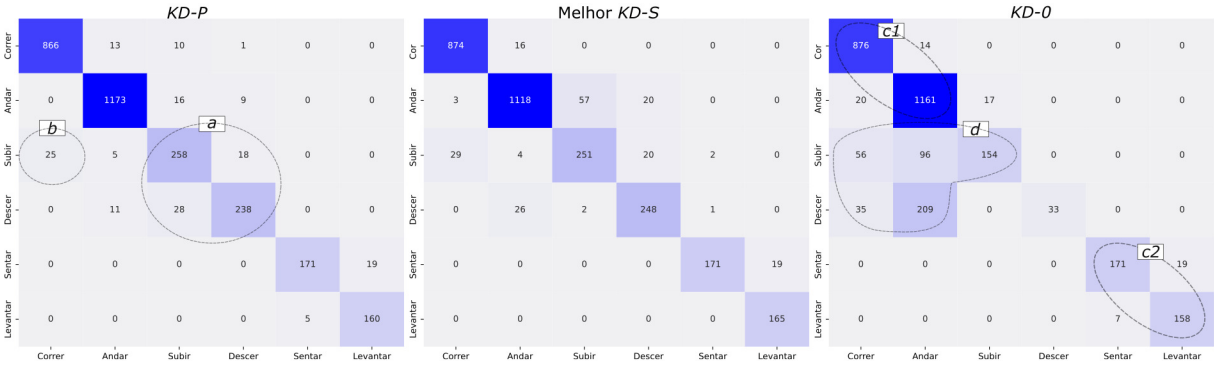


Figura 17 – Matriz de confusão do teste do modelo professor (*KD-P*), melhor modelo estudante destilado (*KD-S*) e do modelo estudante sem destilação (*KD-0*).

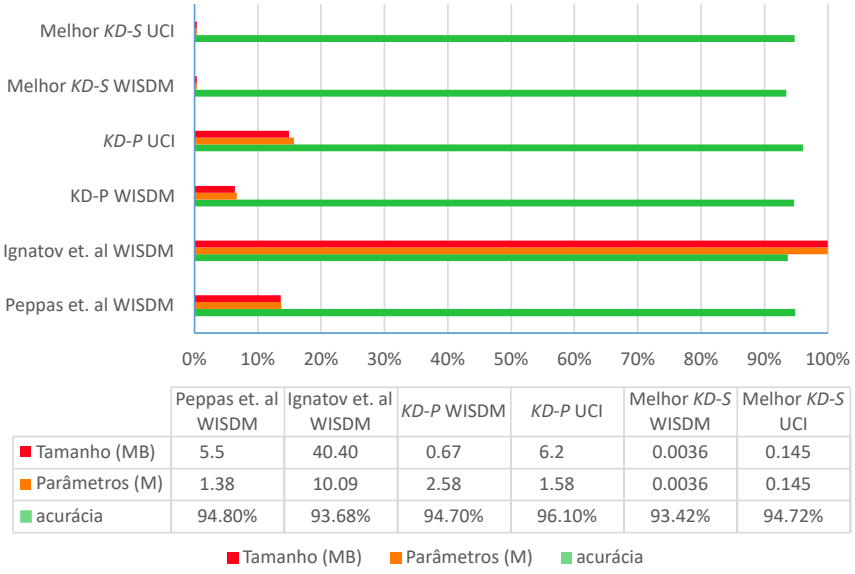
Quando comparamos as matrizes de confusão do modelo *KD-P* e do modelo *KD-S*, a maior confusão está entre as classes subir escadas e descer as escadas (regiões *a* das matrizes). Para as predições destes dois modelos, é interessante observar que a atividade subir escadas é frequentemente confundida com a classe correr (regiões *b* das matrizes).

O modelo *KD-0* é capaz de classificar as atividades correr, andar, sentar e levantar (círculos *c1* e *c2*), mas as atividades subir e descer escadas possuem confusão com andar e correr, em especial atividade com as atividades descer escadas que foi confundida com a

atividade andar (região *d*). Quando comparamos a capacidade de generalização do modelo *KD-S* com o modelo *KD-0*, podemos concluir que destilação de conhecimento atua como um método de regularização durante o treinamento e evita que sejam feitos sobre-ajuste em relação a base de treino, evitando assim que seja inserido viés no modelo.

5.3 Discussão

Os resultados dos experimentos obtidos são sumarizados na Figura 18 em relação ao número de parâmetros do modelo, o tamanho em disco e a acurácia do modelo. Em especial devemos destacar que a compressão do modelo estudante em relação ao modelo professor para a base de dados UCI-RAH foi de 42 vezes e 18.7 vezes para a base de dados WISDM. Os resultados da destilação da base WISDM deve ser destacado, pois obteve uma taxa de compressão de 37 vezes quando comparado ao trabalho de Peppas *et al.* (2020) e 270 vezes quando comparado ao trabalho de Ignatov (2018), enquanto manteve o valor de *F1-Score* superior. Sendo assim, a abordagem proposta foi capaz de executar a tarefa de compressão de modelos professores treinados para RAH enquanto manteve uma acurácia muito próxima do modelo professor.



	Peppas et. al WISDM	Ignatov et. al WISDM	*KD-P* WISDM	*KD-P* UCI	Melhor *KD-S* WISDM	Melhor *KD-S* UCI
Tamanho (MB)	5.5	40.40	0.67	6.2	0.0036	0.145
Parâmetros (M)	1.38	10.09	2.58	1.58	0.0036	0.145
acurácia	94.80%	93.68%	94.70%	96.10%	93.42%	94.72%

Figura 18 – Acurácia, número de paramêtros e espaço em disco para os modelos avaliados.

6 CONCLUSÕES

As redes neurais profundas obtiveram sucesso notável com bom desempenho, especialmente nos cenários do mundo real com dados em grande escala, porque a sobreparametrização melhora o desempenho da generalização quando novos dados são considerados. No entanto, muitos desses modelos baseado em rede neurais se tornam complexos a medida que crescem seu número de parâmetros e a implantação desses modelos profundos em dispositivos móveis e sistemas embarcados é um grande desafio, devido à limitada capacidade computacional e memória dos dispositivos.

Para mitigar essas limitações, o método KD-HAR proposto emprega um mecanismo de destilação de conhecimento para gerar redes compactas enquanto utiliza o conhecimento de uma rede mais complexa para melhorar o desempenho e a capacidade de generalização de uma rede de menor complexidade. Nossos resultados mostram que é possível obter redes adequadas para utilização em dispositivos embarcados através de uma abordagem de destilação de conhecimento, que transfere o aprendizado de um modelo treinado em um contexto sem limitação de recursos para outro modelo, de menor capacidade, mas que pode ser utilizado em dispositivos com recursos computacionais limitados, como é o caso dos *smartwatches*. Em nossos melhores resultados, a acurácia do modelo compacto (estudante) foi superior aos resultados dos trabalhos *baselines* quando levamos em consideração a métrica *F1-Score*.

Nossas descobertas podem servir como orientação geral para o projeto de sistemas de reconhecimento de atividade humana usando sensores de *smartphones*, viabilizando assim, a implantação em dispositivos de baixo custo que estão presentes no cotidiano das pessoas.

Direções futuras podem estar em estender a comparação do método com outras abordagens de compressão de RNP, tal como a utilização de poda de redes neurais ou quantização. Além disso, é possível implementar a união dessas técnicas para a obtenção de modelos com maior taxa de compressão.

Podemos utilizar uma abordagem baseada no conceito de destilação de conhecimento para transferir o conhecimento de modelos professores com dados não rotulados. A utilização de dados não rotulados pode ser uma forma de mitigar a baixa disponibilidade de dados de *datasets* de RAH no contexto móvel e vestível. Além disso, outras técnicas de treinamento semi-supervisionado ou auto-supervisionado podem ser potencialmente incorporadas à estrutura para melhor aproveitar dados não rotulados.

Existe potencial em avaliar a utilização de diferentes representações no treinamento do modelo professor e modelo aluno, habilitando a modalidade cruzada entre tipos de

sensores diferentes. O uso de diferentes representações no modelo professor talvez mantenham características da forma global dos sinais que são perdidas durante as convoluções, aumentando a capacidade do modelo aluno que utiliza apenas dados de sinais brutos.

Também é possível implementar um processo mais robusto de otimização de hiperparâmetros não só para a seleção do modelo professor, mas também para o processo de escolha de arquitetura e configurações de α e da temperatura no processo da destilação, o que permitiria avaliar o processo de compressão como uma otimização multiobjetivo, onde modelos estudantes potenciais seriam avaliados sobre o número de parâmetros (minimação) e acurácia (maximização).

REFERÊNCIAS

ANGUITA, D.; GHIO, A.; ONETO, L.; PARRA, X.; REYES-ORTIZ, J. L. A Public Domain Dataset for Human Activity Recognition using Smartphones. *In*: **21st European Symposium on Artificial Neural Networks**. [*S.l.: s.n.*], 2013. Disponível em: http://hdl.handle.net/2117/20897.

BANOS, O.; GALVEZ, J.-M.; DAMAS, M.; POMARES, H.; ROJAS, I. Window Size Impact in Human Activity Recognition. **Sensors**, MDPI AG, v. 14, n. 4, p. 6474–6499, 2014. Disponível em: https://www.mdpi.com/1424-8220/14/4/6474.

BOEHNER, A. W. A Smartphone Application for a Portable Fall Detection System. *In*: **Proceedings of The National Conference On Undergraduate Research (NCUR)**. [*S.l.: s.n.*], 2013. Disponível em: http://libjournals.unca.edu/ncur/wp-content/uploads/2021/09/423-Boehner.pdf.

BRAGANCA, H. L. d. S. **Reconhecimento de atividades humanas usando medidas estatísticas dos sensores inerciais dos smartphones**. mar. 2019. Dissertação (Dissertação de Mestrado) — Programa de Pós-graduação em Informática, mar. 2019. Instituto de Computação. Disponível em: https://tede.ufam.edu.br/handle/tede/7126.

BRAJDIC, A.; HARLE, R. Walk detection and step counting on unconstrained smartphones. *In*: **Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing**. ACM, 2013. Disponível em: https://dl.acm.org/doi/10.1145/2493432.2493449.

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 Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 54, n. 4, p. 1–40, 2021.

CHEN, Z.; ZHANG, L.; CAO, Z.; GUO, J. Distilling the knowledge from handcrafted features for human activity recognition. **IEEE Transactions on Industrial Informatics**, IEEE, v. 14, n. 10, p. 4334–4342, 2018.

CHENG, W.-Y.; SCOTLAND, A.; LIPSMEIER, F.; KILCHENMANN, T.; JIN, L.; SCHJODT-ERIKSEN, J.; WOLF, D.; ZHANG-SCHAERER, Y.-P.; GARCIA, I. F.; SIEBOURG-POLSTER, J.; SOTO, J.; VERSELIS, L.; MARTIN-FACKLAM, M.; BOESS, F.; KOLLER, M.; GRUNDMAN, M.; MONSCH, A.; POSTUMA, R.; GHOSH, A.; KREMER, T.; TAYLOR, K.; CZECH, C.; GOSSENS, C.; LINDEMANN, M. Human Activity Recognition from Sensor-Based Large-Scale Continuous Monitoring of Parkinson's Disease Patients. *In*: **2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)**. IEEE, 2017. p. 249–250. Disponível em: https://ieeexplore.ieee.org/document/8010642.

CHENG, X.; RAO, Z.; CHEN, Y.; ZHANG, Q. Explaining knowledge distillation by quantifying the knowledge. *In*: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [*S.l.: s.n.*], 2020. p. 12925–12935.

GONG, Y.; LIU, L.; YANG, M.; BOURDEV, L. Compressing deep convolutional networks using vector quantization. **arXiv preprint arXiv:1412.6115**, 2014.

GOU, J.; YU, B.; MAYBANK, S. J.; TAO, D. Knowledge Distillation: A Survey. **International Journal of Computer Vision**, Springer Science and Business Media LLC, v. 129, n. 6, p. 1789–1819, mar 2021. Disponível em: https://link.springer.com/article/10.1007/s11263-021-01453-z.

Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. **arXiv e-prints**, p. arXiv:1503.02531, mar. 2015. Disponível em: https://arxiv.org/abs/1503.02531.

HINTON, G.; VINYALS, O.; DEAN, J. *et al.* Distilling the knowledge in a neural network. **arXiv preprint arXiv:1503.02531**, v. 2, n. 7, 2015.

IGNATOV, A. Real-time human activity recognition from accelerometer data using convolutional neural networks. **Applied Soft Computing**, Elsevier, v. 62, p. 915–922, 2018.

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: Association for Computing Machinery, 2015. (MM '15), p. 1307–1310. ISBN 9781450334594. Disponível em: https://doi.org/10.1145/2733373.2806333.

Jordao, A.; Nazare ANTONIO C., J.; Sena, J.; Robson Schwartz, W. Human Activity Recognition Based on Wearable Sensor Data: A Standardization of the State-of-the-Art. **arXiv e-prints**, 2018. Disponível em: https://arxiv.org/abs/1806.05226.

Karantonis, D. M.; Narayanan, M. R.; Mathie, M.; Lovell, N. H.; Celler, B. G. Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring. **IEEE Transactions on Information Technology in Biomedicine**, Institute of Electrical and Electronics Engineers (IEEE), v. 10, n. 1, p. 156–167, 2006. Disponível em: https://ieeexplore.ieee.org/document/1573717.

KWAPISZ, J. R.; WEISS, G. M.; MOORE, S. A. Activity Recognition Using Cell Phone Accelerometers. **SIGKDD Explor. Newsl.**, Association for Computing Machinery, New York, NY, USA, v. 12, n. 2, p. 74–82, 2011. ISSN 1931-0145. Disponível em: https://dl.acm.org/doi/10.1145/1964897.1964918.

LARA, O. D.; LABRADOR, M. A. A Survey on Human Activity Recognition using Wearable Sensors. **IEEE Communications Surveys and Tutorials**, Institute of Electrical and Electronics Engineers (IEEE), v. 15, n. 3, p. 1192–1209, 2013. Disponível em: https://ieeexplore.ieee.org/document/6365160/.

LEE, Y.-G.; JEONG, W. S.; YOON, G. Smartphone-Based Mobile Health Monitoring. **Telemedicine and e-Health**, v. 18, n. 8, p. 585–590, 2012. PMID: 23061640. Disponível em: https://doi.org/10.1089/tmj.2011.0245.

LI, F.; SHIRAHAMA, K.; NISAR, M. A.; KöPING, L.; GRZEGORZEK, M. Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors. **Sensors**, v. 18, n. 2, 2018. ISSN 1424-8220. Disponível em: https://www.mdpi.com/1424-8220/18/2/679.

Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. **arXiv e-prints**, p. arXiv:1603.06560, mar. 2016. Disponível em: https://arxiv.org/abs/1603.06560.

NAKANO, K.; CHAKRABORTY, B. Effect of dynamic feature for human activity recognition using smartphone sensors. *In*: **2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)**. IEEE, 2017. p. 539–543. Disponível em: https://ieeexplore.ieee.org/document/8256516.

NI, J.; SARBAJNA, R.; LIU, Y.; NGU, A. H.; YAN, Y. Cross-modal knowledge distillation for vision-to-sensor action recognition. *In*: IEEE. **ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [*S.l.: s.n.*], 2022. p. 4448–4452.

NWEKE, H. F.; TEH, Y. W.; MUJTABA, G.; AL-GARADI, M. A. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. **Information Fusion**, v. 46, p. 147–170, 2019. ISSN 1566-2535. Disponível em: https://www.sciencedirect.com/science/article/pii/S1566253518304135.

PEPPAS, K.; TSOLAKIS, A. C.; KRINIDIS, S.; TZOVARAS, D. Real-Time Physical Activity Recognition on Smart Mobile Devices Using Convolutional Neural Networks. **Applied Sciences**, v. 10, n. 23, 2020. ISSN 2076-3417. Disponível em: https://www.mdpi.com/2076-3417/10/23/8482.

POUYANFAR, S.; SADIQ, S.; YAN, Y.; TIAN, H.; TAO, Y.; REYES, M. P.; SHYU, M.-L.; CHEN, S.-C.; IYENGAR, S. S. A Survey on Deep Learning: Algorithms, Techniques, and Applications. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 51, n. 5, 2018. ISSN 0360-0300. Disponível em: https://doi.org/10.1145/3234150.

RAVI, D.; WONG, C.; LO, B.; YANG, G.-Z. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. *In*: IEEE. **2016 IEEE 13th international conference on wearable and implantable body sensor networks (BSN)**. [*S.l.: s.n.*], 2016. p. 71–76.

ROMERO, A.; BALLAS, N.; KAHOU, S. E.; CHASSANG, A.; GATTA, C.; BENGIO, Y. Fitnets: Hints for thin deep nets. **arXiv preprint arXiv:1412.6550**, 2014.

RONAO, C. A.; CHO, S.-B. Human activity recognition with smartphone sensors using deep learning neural networks. **Expert systems with applications**, Elsevier, v. 59, p. 235–244, 2016.

SHOAIB, M.; BOSCH, S.; INCEL, O. D.; SCHOLTEN, H.; HAVINGA, P. J. M. Fusion of Smartphone Motion Sensors for Physical Activity Recognition. **Sensors**, v. 14, n. 6, p. 10146–10176, 2014. ISSN 1424-8220. Disponível em: https://www.mdpi.com/1424-8220/14/6/10146.

SHOAIB, M.; BOSCH, S.; INCEL, O. D.; SCHOLTEN, H.; HAVINGA, P. J. M. A Survey of Online Activity Recognition Using Mobile Phones. **Sensors**, v. 15, n. 1, p. 2059–2085, 2015. ISSN 1424-8220. Disponível em: https://www.mdpi.com/1424-8220/15/1/2059.

SRINIVAS, S.; BABU, R. V. Data-free parameter pruning for deep neural networks. **arXiv preprint arXiv:1507.06149**, 2015.

TANG, C. I.; PEREZ-POZUELO, I.; SPATHIS, D.; BRAGE, S.; WAREHAM, N.; MASCOLO, C. Selfhar: Improving human activity recognition through self-training with unlabeled data. **arXiv preprint arXiv:2102.06073**, 2021.

VAVOULAS, G.; CHATZAKI, C.; MALLIOTAKIS, T.; PEDIADITIS, M.; TSIKNAKIS, M. The mobiact dataset: Recognition of activities of daily living using smartphones. *In*: **International Conference on Information and Communication Technologies for Ageing Well and e-Health**. [*S.l.: s.n.*], 2016. p. 143–151.

VU, D.-Q.; LE, N.; WANG, J.-C. Teaching yourself: A self-knowledge distillation approach to action recognition. **IEEE Access**, IEEE, v. 9, p. 105711–105723, 2021.

WANG, J.; CHEN, Y.; HAO, S.; PENG, X.; HU, L. Deep learning for sensor-based activity recognition: A survey. **Pattern Recognition Letters**, Elsevier, v. 119, p. 3–11, 2019.

WANG, Z.; MENG, F.; YUAN, G.; YAN, Q.; XIA, S. An overview of human activity recognition based on smartphone. **Sensor Review**, Emerald, v. 39, n. 2, p. 288–306, 2019. Disponível em: https://www.emerald.com/insight/content/doi/10.1108/SR-11-2017-0245/.

WILSON, C.; HARGREAVES, T.; HAUXWELL-BALDWIN, R. Smart homes and their users: a systematic analysis and key challenges. **Personal and Ubiquitous Computing**, Springer Science and Business Media LLC, v. 19, n. 2, p. 463–476, 2015. Disponível em: https://link.springer.com/article/10.1007/s00779-014-0813-0.

WU, J.; LENG, C.; WANG, Y.; HU, Q.; CHENG, J. Quantized convolutional neural networks for mobile devices. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [*S.l.: s.n.*], 2016. p. 4820–4828.

Xue, L.; Xiandong, S.; Lanshun, N.; Jiazhen, L.; Renjie, D.; Dechen, Z.; Dianhui, C. Understanding and Improving Deep Neural Network for Activity Recognition. **arXiv e-prints**, p. arXiv:1805.07020, maio 2018. Disponível em: https://arxiv.org/abs/1805.07020.

YANG, J.; LEE, J.; CHOI, J. Activity Recognition Based on RFID Object Usage for Smart Mobile Devices. **Journal of Computer Science and Technology**, Springer Science and Business Media LLC, v. 26, n. 2, p. 239–246, 2011. Disponível em: https://link.springer.com/article/10.1007/s11390-011-9430-9.

YANG, J. B.; NGUYEN, M. N.; SAN, P. P.; LI, X. L.; KRISHNASWAMY, S. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. *In*: **Proceedings of the 24th International Conference on Artificial Intelligence**. AAAI Press, 2015. (IJCAI'15), p. 3995–4001. ISBN 9781577357384. Disponível em: http://ijcai.org/papers15/Papers/IJCAI15-561.pdf.

YIM, J.; JOO, D.; BAE, J.; KIM, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. *In*: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [*S.l.: s.n.*], 2017. p. 4133–4141.

Yu, T.; Zhu, H. Hyper-Parameter Optimization: A Review of Algorithms and Applications. **arXiv e-prints**, 2020. Disponível em: https://arxiv.org/abs/2003.05689.