

UNIVERSIDADE FUMEC
Faculdade de Ciências Empresariais
Programa de Pós-Graduação em Sistemas de Informação e Gestão do
Conhecimento

RUBENS DE OLIVEIRA SANTOS

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA O
RECONHECIMENTO DE ATIVIDADES HUMANAS COM DADOS DE PONTOS DE
REFERÊNCIA DO CORPO HUMANO

Belo Horizonte

2021

RUBENS DE OLIVEIRA SANTOS

**APLICAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA O
RECONHECIMENTO DE ATIVIDADES HUMANAS COM DADOS DE PONTOS DE
REFERÊNCIA DO CORPO HUMANO**

Projeto de dissertação de mestrado apresentado ao Programa de Pós-Graduação em Sistemas de Informação e Gestão do Conhecimento da Universidade FUMEC como parte dos requisitos para a obtenção do título de Mestre em Sistemas de Informação e Gestão do Conhecimento.

Área de concentração: Gestão de Sistemas de Informação e do Conhecimento.

Linha de pesquisa: Tecnologia e Sistemas de Informação.

Orientador: Prof. Dr. Fernando Silva Parreiras

Belo Horizonte

2021

Dados Internacionais de Catalogação na Publicação (CIP)

S237a Santos, Rubens de Oliveira, 1982-

Aplicação de técnicas de aprendizado de máquina para o reconhecimento de atividades humanas com dados de pontos de referência do corpo humano / Rubens de Oliveira Santos. - Belo Horizonte, 2021.

70 f.: il.

Orientador: Fernando Silva Parreiras

Dissertação (Mestrado em Sistemas de Informação e Gestão do Conhecimento), Universidade FUMEC, Faculdade de Ciências Empresariais, Belo Horizonte, 2021.

1. Aprendizado do computador. 2. Human Activity Recognition. 3. Redes Neurais (Computação). 4. Processamento eletrônico de dados. I. Título. II. Parreiras, Fernando Silva. III. Universidade FUMEC, Faculdade de Ciências Empresariais.

CDU: 004.8



Dissertação intitulada “**Aplicação de Técnicas de Aprendizado de Máquina para o Reconhecimento de Atividades Humanas Com Dados de Pontos de Referência do Corpo Humano**” de autoria de Rubens de Oliveira Santos, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Fernando Silva Parreiras – Universidade FUMEC
(Orientador)

Prof. Dr. João Victor Boechat Gomide – Universidade FUMEC
(Examinador Interno)

Prof. Dr. Eric de Paula Ferreira – UEMG
(Examinador Externo)

Prof. Dr. Fernando Silva Parreiras
Coordenador do Programa de Pós-Graduação em Sistemas de Informação e Gestão do
Conhecimento da Universidade FUMEC

Belo Horizonte, 21 de maio de 2021.

Fernanda Silva Parreiras

Professor João Victor Boechat Gomide Eric de Paula Ferreira

	TITLE
	FILE NAME
	REQUEST ID
REQUESTED	REQUESTED BY
	STATUS ● Completed

Professor (fernando.parreiras@fumeec.br)


	18/07/2021 03:37:55UTC±0		18/07/2021 03:38:02UTC±0 187.111.30.10
SENDED		SIGNED	

Professor (jvictor@fumeec.br)

	19/07/2021 14:03:47UTC±0		19/07/2021 14:03:56UTC±0 186.214.223.167
SENDED		SIGNED	

Professor (eric.p.f@gmail.com)

	19/07/2021 17:47:22UTC±0		19/07/2021 17:47:41UTC±0 131.161.13.2
SENDED		SIGNED	

	19/07/2021 17:47:41 UTC±0	The document has been completed.	
COMPLETED			

AGRADECIMENTOS

Gostaria de agradecer a toda a minha família, principalmente às minhas amadas filha e esposa por todo o apoio e paciência.

Ao meu pai e aos meus irmãos, pelo apoio e compreensão. À minha mãe, por suas orações e palavras de apoio.

Imensamente ao meu orientador, Dr. Fernando Silva Parreiras, pelos ensinamentos, pelas orientações e pela oportunidade.

Aos colegas do LAIS (Laboratório de Sistemas de Informação Avançada), principalmente a Daniel Henrique Mourão Falci e a Marco Antônio, cada um deles tendo sua parcela de contribuição.

Por fim, devo expressar minha gratidão aos alunos e ao coordenador da UniverCemig pela paciência e pelas sugestões durante a coleta de dados realizada.

RESUMO

A adoção de métodos de aprendizado de máquina para reconhecimento de atividades humanas, tem apresentado notáveis resultados na extração de características discriminativas de sequências adquiridas por intermédio de imagens. Realizar o reconhecimento de atividades executadas pelos alunos em treinamento com a utilização de técnicas de aprendizado de máquina pode auxiliar a UniverCemig no processo de certificação de aprendizado dos alunos. Este projeto de pesquisa analisa quais técnicas de aprendizado de máquina podem ser utilizadas para realizar o reconhecimento de atividades humanas na formação do eletrecista na UniverCemig. Assim, dar-se-á o passo inicial nesse sentido, com a coleta de dados e com a implementação de algoritmos de aprendizado de máquina, K-Nearest Neighbors(KNN) e Support Vector Machine (SVM), atingindo 91% e 96% de performance respectivamente e de aprendizado de máquina profundo, Gated Recurrent Unit (GRU) e Long Short Term Memory (LSTM), com performance de 96% em ambos para realizar o reconhecimento de atividades humanas com dados obtidos dos pontos de junção do esqueleto.

Palavras-chave: Reconhecimento de Atividade Humana, Aprendizado de Máquina, Aprendizagem de Máquina Profundo.

ABSTRACT

The adoption of machine learning methods for the recognition of human activities, has shown remarkable results in the extraction of discriminative characteristics from sequences acquired through images. Recognizing activities performed by students in training using machine learning techniques can assist UniverCemig in the student learning certification process. This research project analyzes which machine learning techniques can be used to perform the recognition of human activities in the training of electricians at UniverCemig. Thus, the initial step in this direction will be taken, with the data collected and implementation of machine learning algorithms, K-Nearest Neighbors (KNN) and Support Vector Machine (SVM), reaching 91% and 96% performance respectively and deep machine learning, Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM), with 96% performance on both to perform recognition of human activities with data obtained from the junction points of the skeleton.

Keywords: Human Activity Recognition, Machine Learning, Deep Machine Learning.

LISTA DE FIGURAS

Figura 1 – Definição de Reconhecimento de Atividade Humana	20
Figura 2 – Modelos de Janelas Deslizantes	21
Figura 3 – Pontos de Referência do Corpo Humano	22
Figura 4 – Modelo Linear	24
Figura 5 – Modelo Linearmente Separável	25
Figura 6 – Modelo Linearmente Separável e Não Separável.....	26
Figura 7 – Modelo de Neurônio	28
Figura 8 – Modelo de Arquitetural FNN.....	30
Figura 9 – A Estrutura Básica de Redes Neurais Recorrentes.....	30
Figura 10 – Recurrent Neural Networks (RNN) com Duas Camadas Densas	31
Figura 11 – Modelo de Arquitetural Recurrent Neural Networks (LSTM)	32
Figura 12 – Arquitetura Gated Recurrent Unit.....	34
Figura 13 – Resultado da aplicação da camada convolucional (CNN)	36
Figura 14 – Modelo de Arquitetural Recurrent Neural Networks (LSTM)	37
Figura 15 – Proposta de Modelo de Execução do Projeto	42
Figura 16 – Fluxo de coleta dos dados	44
Figura 17 – Pontos de Referência do Corpo Humano.....	50
Figura 18 – Mapa de calor representado o ponto-chave pulso direito.....	51
Figura 19 – Gráfico com resultados por atividade.....	52
Figura 20 – Gráfico de acuracia de KNN e SVM por janela de tempo	56
Figura 21 – Modelo de Treino Gated Recurrent Unit.....	57
Figura 22 – Modelo de Treino Long Short Term Memory.....	59
Figura 23 – Modelo de Arquitetura Híbrida	62

LISTA DE TABELAS

Tabela 1 – Passos para alcançar os objetivos deste trabalho	41
Tabela 2 – Hiperparâmetros KNN	54
Tabela 3 – Hiperparâmetros de Configuração da Técnicas SVM.....	54
Tabela 4 – Resultados das Técnicas de SVM e KNN	55
Tabela 5 – Configuração Modelo GRU e Hiperparâmetros.....	58
Tabela 6 – Configuração Modelo LSTM e Hiperparâmetros	59
Tabela 7 – Resultados das Técnicas de SVM e KNN	60
Tabela 8 – Modelo de <i>framework</i> proposto por Gumaei et al. (2019)	61

SIGLAS

- AM** Aprendizado de Máquina. 12-14, 22, 23, 26-28, 38, 39, 44, 45, 47, 48, 52
- AMP** Aprendizado de Máquina Profundo. 12-14, 27, 44, 49, 53, 52, 55
- CNN** Convolutional Neural Network. 35, 36, 38, 48, 49
- FNN** Feedforward Neural Network. 28-30, 35
- GRU** Gated Recurrent Unit. 12, 13, 33, 34, 38, 39, 44, 56-61
- KNN** K-Nearest Neighbors. 13, 26, 27, 39, 44, 52–55, 59, 61
- LSTM** Long Short Term Memory. 12, 13, 29, 32–34, 37-39, 44, 57-61
- RAH** Reconhecimento de Atividades Humanas. 12, 13, 19–22, 28, 38, 39, 41, 44, 47, 50, 55
- RNA** Redes Neurais Artificiais. 27–30, 35
- RNN** Redes Neurais Recorrentes. 12, 29–33
- SVM** Support Vector Machine. 13, 23, 24, 26, 38, 39, 44, 52–55, 59, 61
- URS** Unidade Recorrente Simples. 38

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Problema de Pesquisa	14
1.2 Objetivos	16
1.2.1 Objetivo Geral.....	16
1.2.2 Objetivos Específicos	16
1.3 Aderência ao Programa de Pós-Graduação da FUMEC em Sistemas de Informação e Gestão do Conhecimento	17
2 FUNDAMENTAÇÃO TEÓRICA	19
2.1 Visão Computacional	19
2.2 Reconhecimento de Atividade Humana.....	19
2.2.1 Definição de Reconhecimento de Atividades Humanas.....	20
2.2.2 Janelas Deslizantes	21
2.3 Pontos de Referência do corpo humano.....	22
2.4 Aprendizado de Máquina	22
2.4.1 Support Vector Machine.....	23
2.4.1.1 Máquina de Vetor de Suporte Linearmente Separável	24
2.4.1.2 Máquina de Vetor de Suporte Não Linearmente Separável	25
2.4.2 K-Nearest Neighbors.....	26
2.4.3 Aprendizado de Máquina Profundo.....	27
2.4.3.1 Redes Neurais Artificiais.....	27
2.4.3.2 Feedforward Neural Network.....	28
2.4.3.3 Redes Neurais Recorrentes	30
2.4.3.4 Long Short Term Memory	32
2.4.3.5 Gated Recurrent Unit.....	33
2.4.3.6 Convolutional Neural Network	35
3 TRABALHOS RELACIONADOS	38

3.1 Técnicas de Machine Learning	38
4 METODOLOGIA	40
4.1 Caracterização da Pesquisa	40
4.2 Paradigma da Pesquisa.....	40
4.3 Estratégias da Pesquisa	41
4.4 Proposta de Modelo.....	42
4.4.1 Identificar Atividades executadas.....	42
4.4.2 Coleta de dados.....	43
4.4.2.1 Identificação dos Participantes.....	44
4.4.2.2 Filmagem do Treinamento.....	44
4.4.2.3 Segmentação do Vídeo e Geração de Imagens.....	45
4.4.2.4 Extração de Pontos de Referência do Corpo Humano.....	45
4.4.2.5 Anotação da Atividade.....	45
4.4.2.6 Artefato Gerado.....	45
4.4.3 Aplicar Técnicas de Aprendizado de Máquina	45
4.4.4 Análise dos Resultados.....	46
4.4.4.1 Acurácia.....	46
4.4.4.2 Precisão.....	47
4.4.4.3 Revocação.....	47
5 COLETA DE DADOS.....	48
5.1 Abordagem	48
5.1.1 Filmagem do Treinamento	48
5.1.2 Segmentação do Vídeo, Geração de Imagens e Filtragem	48
5.1.3 Pontos de Referência do Corpo Humano	49
5.1.4 Processo Manual de Anotação dos dados.....	51
5.1.5 Conjunto de dados.....	51

6 RESULTADOS.....	53
6.1 Experimento Aprendizado de Máquina	53
6.1.1 Hiperparâmetros K-Nearest Neighbors	53
6.1.2 Hiperparâmetros Support Vector Machine	54
6.1.3 Análise dos Resultados de Aprendizado de Máquina.....	54
6.2 Experimento Aprendizado de Máquina Profundo	56
6.2.1 Gated Recurrent Unit	57
6.2.2 Long Short Term Memory	58
6.2.3 Análise dos Resultados Aprendizado de Máquina Profundo	59
6.2.3.1 Discussão dos Resultados Aprendizado de Máquina Profundo GRU...61	
6.2.3.2 Discussão dos Resultados Aprendizado de Máquina Profundo LSTM .62	
7 CONCLUSÃO	64
BIBLIOGRAFIA	65

1 INTRODUÇÃO

O reconhecimento da atividade humana (RAH), é um campo de pesquisa ativo em visão computacional. Seu foco principal é detectar automaticamente informações sobre atividades executadas por pessoas através de dispositivos digitais (POPPE, 2010). Com o avanço da tecnologia, diversos dispositivos como: câmera RGB, câmera de profundidade, câmera RGB-D e Microsoft Kinect, foram criados, possibilitando a extração de diferentes informações, e entre essas, pode-se destacar os pontos de referências do corpo humano (TASNIM *et al.*, 2020).

A extração de pontos de referências do corpo humano é uma técnica que tem sido aplicada dentro da visão computacional para realizar o (RAH), estimar poses de um ser humano (STRAKA *et al.*, 2011), identificar o comportamento humano (SAPINSKI *et al.*, 2019) e detecção de expressão facial (FILNTISIS *et al.*, 2019) são exemplos de aplicação dessa técnica.

Pesquisas recentes tem aplicado técnicas de aprendizagem de máquina (AM) e aprendizagem de máquina profunda (AMP) para realizar o RAH sobre conjunto de dados extraídos a partir de pontos referências do corpo humano, como: Tasnim et al. (2020) aplicando Redes Neurais Recorrentes (RNN) em um conjunto de dados com características discriminativas extraídas de pontos de junção do esqueleto, enquanto Yan *et al.* (2018) aplicou técnicas de AM, utilizando modelo de dados públicos como conjunto de dados de ação humana multimodal UTD e jogos 3D. Outras abordagens de AMP têm sido utilizadas, como: Chung et al. (2014) utilizando Gated Recurrent Unit (GRU) e LeCun *et al.* (2015) utilizando Long Short Term Memory (LSTM).

Com a aplicação de técnicas como: AM e AMP, as atividades executadas pelos alunos podem ser automaticamente reconhecidas, auxiliando no processo de treinamento e certificação do aluno, no qual os dados das atividades executadas podem ser extraídos após o treinamento. A adoção de instrumentos automatizados que executem tarefas como RAH contribui para um processo de certificação célere. O referido projeto de pesquisa realizará duas abordagens das técnicas de AM e AMP. A primeira abordagem será implementar técnicas de AM como KNN e SVM. A segunda abordagem será aplicar abordagens de aprendizado de máquina profundo GRU e LSTM.

Os dados para treinamento do modelo de AM e AMP foram extraídos dos

pontos de referência do corpo humano. Este processo se deu a partir de uma filmagem, que foi feita no campo de treinamento de eletricitistas oferecido pela UniverCemig. Durante esta etapa, um passo relevante realizado foi a anotação dos dados, que foi feita através de abordagem supervisionada. Segundo Huynh et al. (2008), a anotação supervisionada se caracteriza pela existência de uma pessoa responsável por verificar a atividade executada e realizar a anotação desta. Para o objetivo de pesquisa, este trabalho foi dividido em três passos: (a) criar um dataset (conjunto de dados), (b) treinar os modelos de AM e AMP para classificação de RAH e (c) comparar os resultados dos algoritmos citados anteriormente.

O treinamento do eletricitista é um processo que inclui diversas tarefas. Porém, o referido trabalho abordará apenas atividade de montagem de estrutura B1. Esta atividade inclui passos como: colocar braceletes no poste, fixar o bastão no poste e fixar pinos no bastão. Esta pesquisa se limita à abordagem proposta, e espera-se obter resultado com as atividades citadas anteriormente aplicando as técnicas de AM e AMP.

1.1 Problema de Pesquisa

A UniverCemig (Universidade da Companhia de Energia Elétrica de Minas Gerais), oferece cursos para o processo de formação e desenvolvimento de competências dos profissionais que atuam no setor elétrico. Estima-se que um treinamento tenha, em média, a duração de 4-6 meses. Com custo e tempo elevados, faz-se necessário criar estratégias embasadas em tecnologias que contribuem para acelerar o processo de aprendizagem, reduzir custos e conseqüentemente agregar de forma positiva a didática deste processo.

Em meio à busca de mecanismos para aprimorar cada vez mais o aprendizado do aluno, a UniverCemig criou o Projeto de Pesquisa e Desenvolvimento D0595 – Desenvolvimento de Conhecimento Tácito e Alternância Pedagógica na Formação de Profissionais para atuação no Setor Elétrico. A proposta desse projeto é adicionar diferentes tecnologias em todo o ciclo do processo de aprendizagem do eletricitista.

O processo de aprendizagem constitui-se em numerosas etapas, incluindo ambiente simulado por meio da realidade aumentada e gamificação, bem como aulas teóricas. Após passar por todas etapas citadas anteriormente, o aluno é encaminhado para o treinamento em campo, onde deverá viver situações reais do dia a dia de um profissional do setor elétrico.

Com base nas informações supracitadas, dentro do processo proposto pela UniverCemig e durante o treinamento em campo é necessário automatizar o reconhecimento das atividades executadas pelos alunos de forma a facilitar a análise do especialista, que conduz o treinamento e necessita observar em tempo real as atividades desempenhadas por cada aluno.

Diante disto, a automatização irá extinguir a exigência pela presença em tempo real do instrutor para aferir a execução das atividades pertinentes ao processo, contribuindo para a redução do período de duração do curso e, conseqüentemente, para atenuar seu valor de investimento. Além disso, os alunos terão acrescido ao processo de aprendizagem o acesso a relatórios que, oriundos da automatização, detalham suas ações, contribuindo para uma auto análise e para o aprimoramento da execução das tarefas. Isso leva à seguinte questão:

Quais técnicas de aprendizado de máquina devem ser utilizadas a partir da

**iteratura para realizar o reconhecimento de atividades humanas na formação do
eletricista na UniverCemig?**

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Identificar técnicas de aprendizado de máquina para realizar o reconhecimento de atividades humanas com dados obtidos de imagens a partir de atividades executadas por um ser humano.

1.2.2 Objetivos Específicos

1. Criar um conjunto de dados (*dataset*) com as informações obtidas de imagens.
2. Implementar técnicas de aprendizado de máquina e aprendizado de máquina profundo no reconhecimento de atividades humanas.
3. Analisar os resultados das técnicas implementadas.

1.3 Aderência ao Programa de Pós-Graduação da FUMEC em Sistemas de Informação e Gestão do Conhecimento

O programa de pós-graduação da FUMEC em Sistemas de Informação e Gerenciamento de Conhecimento está focado no conhecimento acadêmico, desenvolvimento científico e pesquisa aplicada nas áreas de Sistemas de Informação e Gerenciamento de Conhecimento. O programa está organizado em dois fluxos principais: Tecnologia e Sistemas de Informação e Gerenciamento de Informação e Conhecimento. A abordagem multidisciplinar é um conceito-chave no programa de pós-graduação da FUMEC.

Este trabalho posiciona-se na linha de pesquisa "Tecnologia e Sistemas de Informação", sendo seu objetivo implementação, análise e comparação dos resultados das técnicas de Aprendizado de Máquina e Aprendizado de Máquina Profundo no reconhecimento de atividades humanas. Essas técnicas atuam no contexto da Inteligência Artificial, mais especificamente em Aprendizado de Máquina, suas derivações, ferramentas de implementação e metodologias adequadas. Como há a proposta de modelos de Aprendizado de Máquina e Aprendizado de Máquina Profundo que possibilitem o reconhecimento de atividades humanas, entende-se que o trabalho se enquadra ao Programa de Mestrado da FUMEC.

1.4 Estrutura do documento

Este projeto de dissertação está estruturado em 6 capítulos. O capítulo 1 introduz a temática a ser tratada. O capítulo 2 fornece um panorama superficial sobre conceitos importantes para a contextualização do trabalho. O capítulo 3 aborda os trabalhos relacionados que tiveram como foco de sua pesquisa o reconhecimento de atividades humanas. O capítulo 4 descreve os métodos e os materiais a serem aplicados. O capítulo 5 aborda o processo de coleta de dados. O capítulo 6 apresenta os resultados e as configurações das técnicas implementadas e, por fim, o capítulo 7 apresenta a conclusão.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresenta-se os conceitos centrais utilizados na presente de pesquisa para esclarecer a terminologia e facilitar o entendimento dos tópicos. São eles: Visão Computacional, Reconhecimento de Atividade Humana, Pontos de Junção do Esqueleto e Aprendizado de Máquina.

2.1 Visão Computacional

Visão Computacional é a área da ciência que estuda e procura desenvolver teorias e métodos voltados à extração automática de informações úteis contidas em imagens. Tais imagens são capturadas por dispositivos como câmera de vídeo e scanner (James, L. Crowley, 1995).

As tecnologias que simulam a visão humana como visão computacional são tópicos de pesquisa atuais em constante desenvolvimento, por meio das quais é possível a medição óptica com precisão computacional através da expansão dos sentidos, nutrindo o sistema computacional com uma gama de informações. Este processo é formado por algumas etapas que compõem o sistema de visão computacional, variando conforme aplicações e objetivos originando-se por meio da aquisição de imagem, pré-processamento, extração de características (James L. Crowley, 1995).

A visão computacional é uma subárea da inteligência artificial, com o objetivo de buscar um corpo de conhecimentos que modele artificialmente a visão humana, realizando ações por meio de suas suas funções através de desenvolvimento de softwares e hardwares avançados. Com a utilização de sensores, computadores e algoritmos de aprendizado de máquina, é possível simular a percepção de um ser humano, incluindo ver, sentir, receber dados e entender o que está acontecendo, extraindo informações de tal forma que possa ser aplicado em diversos processos (Tuceryan and Jain, 2015).

2.2 Reconhecimento de Atividade Humana

O Reconhecimento de Atividades Humanas (RAH) é a tarefa de identificar ou prever atividades humanas baseando-se em dados de sensores (Dehghani et al., 2019)

ou em captura de imagens (Vahora and Chauhan, 2019), podendo utilizar técnicas aprendizado de máquina para realizar a inferência. A Figura 1 apresenta um fluxo de RAH.

Figura 1 – Definição de Reconhecimento de Atividade Humana



Figura 1 - Representação de atividade. (1) Atividade sendo executada, (2) Dispositivo de coleta de dados e armazenamento e (3) Processamento dos dados com técnicas de aprendizado de máquina para realizar a inferência da atividade executada. Fonte: (Rosati et al, 2018)

Os dados adquiridos a partir de movimentos executados por um usuário são capturados e convertidos em padrões característicos por meio de uma combinação de componentes de *hardware* e *software*. Após a captura dos dados de observações, é feita a inferência da atividade executada (Rosati et al, 2018). As seções posteriores reveem esses aspectos.

2.2.1 Definição de Reconhecimento de Atividades Humanas

De acordo com Lara and Labrador (2013), Reconhecimento de Atividades Humanas (RAH) é um problema de séries temporais. Os dados utilizados referem-se ao movimento executado pelo usuário em um certo momento. Sendo assim, RAH pode ser abordado das seguintes maneiras: (1) abordagem de séries temporais sequenciais e (2) abordagem de séries temporais com janelas deslizantes.

A primeira definição para séries temporais sequenciais pode ser dada a partir de um conjunto de dados $S_0 \dots S_k - 1$, onde k é definido por uma série temporal e onde cada característica está contida dentro de um intervalo de tempo $I = [t\alpha, t\omega]$, o objetivo é olhar a partição do tempo $\langle I_0 \dots I_r - 1 \rangle$ de I de acordo com os dados no conjunto S e encontrar as classes que representam a atividade executada. Os dados são consecutivos, não podem ser vazios e não se sobrepõe. Essa definição implica que uma pessoa não executa duas atividades ao mesmo tempo (Lara and Labrador,

2013).

A segunda abordagem de séries temporais com janelas deslizantes pode ser dada pelo conjunto de dados $W = W_0 \dots W_{m-1}$ para m sendo o tamanho da janela deslizante, total ou parcialmente rotulada e para cada W_i contendo um conjunto de séries temporais, dado $S = S_0 \dots S_{k-1}$ e um conjunto $A = a_0 \dots a_n - 1$ de atividades rotuladas em que o objetivo é uma função de mapeamento $f: S_i \rightarrow A$, que pode ser avaliado para todos os valores possíveis de S_i . Desse modo, $f(S_i)$ seria o mais semelhante possível à atividade real realizada durante W_i (Lara and Labrador, 2013)

2.2.2 Janelas Deslizantes

Técnicas de janelas deslizantes são amplamente utilizadas ao realizar o RAH por possibilitar a segmentação dos dados de entrada, quebrando-os em janelas de tamanho fixo. Durante a fase de segmentação, os dados capturados dentro de cada janela representam uma aproximação da atividade executada e são levados ao classificador para realizar a inferência. Tais informações são capturadas sob dois conceitos: sobreposição e sem sobreposição. Em janelas sem sobreposição, os dados capturados em um intervalo de tempo não são sobrepostos, ou seja, dados de uma atividade executada não sobrepõem os de outra. Por outro lado, em janelas com sobreposição, dados de outra atividade sendo executada são sobrepostos (Dehghani et al., 2019). A Figura 2 ilustra janelas deslizantes com sobreposição e sem sobreposição de sinais.

Figura 2 – Modelos de Janelas Deslizantes

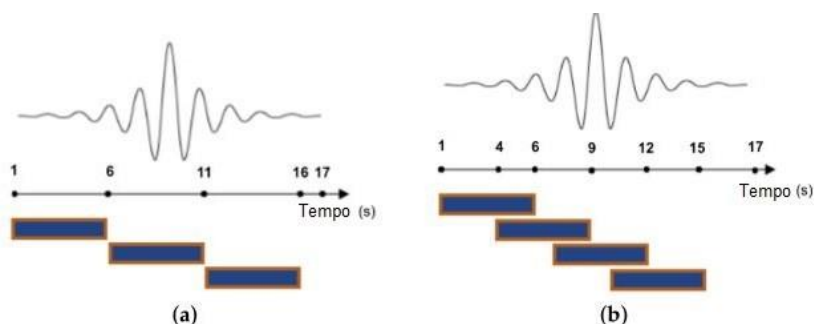


Figura 2 - Janelas Deslizantes. (a) Sem sobreposição dos sinais (b) Com sobreposição dos sinais. Fonte: (Dehghani et al., 2019)

2.3 Pontos de Referência do corpo humano

Um movimento humano é coordenado por diferentes partes do corpo e, com a utilização de tecnologia, é possível realizar sua representação através de coordenadas adquiridas de cada ponto do corpo humano que, em um conjunto de dados, representam uma ação executada, conforme Figura 3. Tal representação pode ser definida como um conjunto de coordenadas, que são utilizadas para descrever a pose de uma pessoa em um determinado espaço temporal (Manzi et al., 2017).

Figura 3 – Pontos de Referência do Corpo Humano

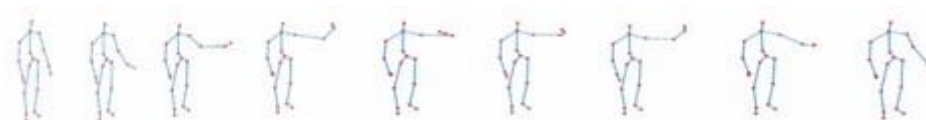


Figura 3 - Representação dos pontos de referência do corpo humano, usados para gerar o conjunto de dados. Fonte: (Manzi et al., 2017)

Pontos chave que compõem um corpo humano podem ser extraídos por câmera de profundidade com rastreador específico ou utilizando outro modelo que faz o mapeamento da imagem. O modelo do esqueleto humano possui algumas variedades, sendo de acordo com o *software* e com o dispositivo adotado. Um modelo esquelético pode ser modelado com vários pontos de junção (Shotton et al., 2011), sendo que cada ponto descreve coordenadas em relação a posição de cada membro (Manzi et al., 2017).

Modelos de dados podem ser criados a partir dos pontos de junção extraídos do modelo esquelético para formar um dataset que possa ser utilizado por técnicas de aprendizado de máquina para realizar o RAH (Tasnim et al., 2020).

2.4 Aprendizado de Máquina

Aprendizado de Máquina (AM) é um subcampo da inteligência artificial que visa explorar modelos de dados e construir algoritmos que possibilitem compreender os dados de maneira autônoma. Utilizando algoritmo de AM, é possível pegar um

conjunto de dados de entrada e com base em determinados padrões encontrados inferir as saídas. Para Sugiyama and Kawanabe (2012), o principal propósito de AM é a construção de sistemas que podem se adaptar em ambientes com constantes mudanças e aprender com sua experiência. Outra definição, trazida por Mitchell et al. (2013), reforça o que foi dito anteriormente, caracterizando o AM como um sistema de computador criado para otimizar a performance usando exemplo de dados e experiência passada. O AM pode ser dividido em várias categorias; abaixo, explica-se as mais comuns:

- **Aprendizagem Supervisionada:** Nesta categoria, o AM recebe todos os dados de entrada rotulados como um conjunto de treinamento e faz previsões sobre dados não vistos anteriormente. Uma desvantagem dessa categoria é que, na maioria das vezes, os dados são rotulados manualmente para orientar o processo de aprendizado (Davila et al., 2017).
- **Aprendizagem Não Supervisionada:** Ao contrário do método supervisionado, esta categoria não conhece os dados de saída (Mitchell et al., 2013).
- **Aprendizado semi-supervisionado:** Esta categoria usa tanto dados rotulados quanto não rotulados. A vantagem dessa categoria é o fato de que apenas um número limitado de amostras rotuladas é necessário (Davila et al., 2017).

2.4.1 Support Vector Machine

De acordo com Rosati et al. (2018), Support Vector Machine é um classificador binário, capaz de diferenciar dados entre duas classes. Sua estrutura é fundamentada na teoria estatística, tendo como objetivo propor técnicas de AM para maximizar a capacidade de generalização do modelo e minimizar o risco estrutural. Para realizar a classificação, os dados são projetados em um espaço multidimensional, possibilitando realizar a classificação entre duas classes. Nesse espaço, a separação entre as classes é obtida com um hiperplano que maximiza sua distância dos chamados vetores de suporte. De acordo com Janidarmian et al. (1995), SVM consiste em conjuntos de modelos supervisionados de aprendizagem e algoritmos associados que analisam dados usados para problemas classificação e regressão. A Figura 4 ilustra um modelo linear projetado em um espaço multidimensional.

Figura 4 – Modelo Linear

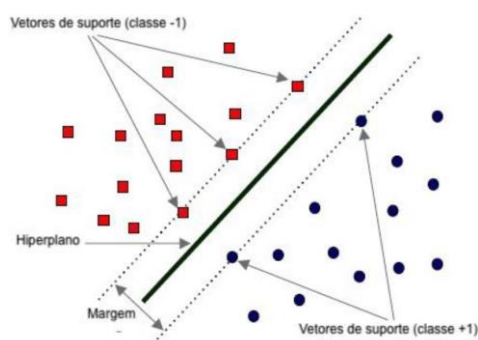


Figura 4 - Modelo linear separado por sua margem em um hiperplano. Fonte: Janidarmian et al. (1995)

Na figura 4, é possível observar que há amostras de classes distintas em cada extremo da margem de separação. Estes dados situados nas fronteiras são chamados de vetores de suporte, o que dá nome também à técnica SVM.

Janidarmian et al. (1995) é capaz de construir, a partir de um conjunto de dados, um modelo capaz de aprender com novos exemplos, atribuindo novas classes aos novos exemplos, com o objetivo de separar categorias distintas por uma margem, sendo esta a mais larga possível (como mostra a Figura 4) no intuito de minimizar o erro de generalização do classificador (Rosati et al., 2018). Este modelo trabalha com duas perspectivas: a classificação de classes linearmente separáveis e não separáveis, que serão apresentadas nas seções posteriores.

2.4.1.1 Máquina de Vetor de Suporte Linearmente Separável

Um conjunto de dados pode ser considerado linearmente separável quando seus dados podem ser separados em um plano multidimensional. Sendo assim, quando Janidarmian et al. (1995) encontra um hiperplano que separa perfeitamente os dados de cada classe e cuja margem de separação seja a maior possível, este é denominado de hiperplano ótimo. Margem é a região delimitada por um par de hiperplanos paralelos que separam as duas classes de dados com a maior distância possível. A Figura 5 ilustra um modelo Janidarmian et al. (1995) linearmente separável. A linha marcada com negrito representa a margem.

Figura 5 – Modelo Linearmente Separável

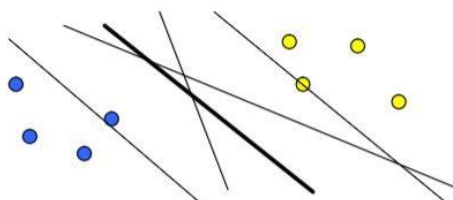


Figura 5 - Modelo linear com duas classes separados por sua margem. Fonte: Janidarmian et al. (1995).

A equação que separa os padrões através de hiperplanos é definida por:

$$w^t \cdot x + b = 0$$

no qual $w^t \cdot x$ é o produto escalar entre os vetores w e x , sendo x o vetor de entrada, w o vetor de pesos e b o bias.

2.4.1.2 Máquina de Vetor de Suporte Não Linearmente Separável

Quando os dados de treinamento são linearmente separáveis, é possível encontrar um hiperplano de margem rígida que atenda ao requisito de separabilidade máxima desde que não haja erros e/ou *outliers* nos dados. Entretanto, caso os dados não sejam linearmente separáveis e/ou deseja-se permitir certa tolerância de erros de classificação (visando evitar *overfitting*), deve-se então encontrar um hiperplano de margem suave, que não se ajustará totalmente e apresentará capacidade de generalização potencialmente superior. Nesse caso, é necessário encontrar um hiperplano que minimize a probabilidade de erro de classificação. A Figura 6 apresenta dois modelos de dados (Rosati et al., 2018).

Figura 6 – Modelo Linearmente Separável e Não Separável

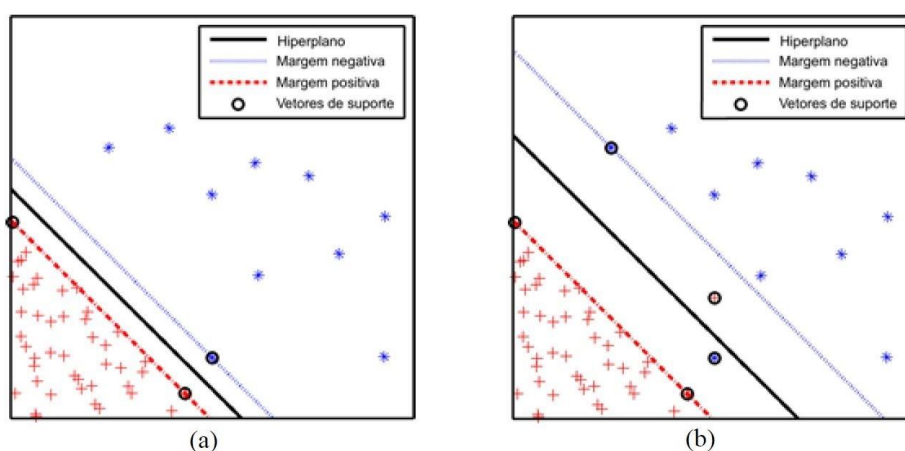


Figura 6 - Modelos SVM de margem rígida (a) e margem suave (b) separados por um hiperplano. Fonte: Rosati et al. (2018)

Na o primeiro modelo (SVM de margem rígida) ilustrado na Figura 6, os dados são linearmente separáveis; neste modelo, é possível observar a separação dos dados. Já no segundo modelo (SVM de margem suave), os dados não foram separados, pois existe um ponto dentro da margem de separação. Isso faz com que SVM aplique técnicas de separação dos dados do tipo *kernel*.

Segundo Rosati et al. (2018), *Kernel* é uma função que recebe os parâmetros de entrada fazo cálculo do produto escalar, sendo definida por:

$$w = \sum_{i,j=1}^N \alpha_i (d_i) \varphi^t(x_i) \cdot \alpha (x_j)$$

onde o vetor de característica α_i corresponde ao padrão de entrada x . As funções que podem ser utilizadas são: Polinomial, Radial Basis Função e Sigmoidal.

2.4.2 K-Nearest Neighbors

De acordo com Sáez et al. (2016), K-Nearest Neighbors (KNN) é um algoritmo de AM para classificação e regressão. O método baseia-se na premissa de que exemplos com características similares devem manter-se próximos. Quando um novo exemplo é adicionado, KNN realiza uma análise de vizinhos mais próximos através do cálculo de distância para classificar uma instância desconhecida (Aguileta et al., 2019). Durante a classificação, KNN analisa a classe de menor custo; caso exista uma classe

com o mesmo menor custo, então é realizado um desempate aleatório entre os grupos de menor custo Janidarmian et al. (2017). A fórmula de início de KNN é:

$$left K_i n = \sqrt{n}$$

onde n é o número de amostras dos dados de treino. Para realizar o cálculo de distância de uma nova amostra, podem ser usadas as distâncias Euclidiana, Cosseno e Minkowski Janidarmian et al. (2017).

2.4.3 Aprendizado de Máquina Profundo

Mitchell et al. (2013) O aprendizado de máquina profundo (AMP) pode ser definido como uma extensão ou subcategoria de AM. Abordagens de aprendizado supervisionado e não supervisionado citadas no item 2.4 também fazem parte do AMP, que foi inspirado por redes neurais artificiais RNA. Nas próximas seções, explicaremos os principais temas.

2.4.3.1 Redes Neurais Artificiais

O artigo publicado por McCulloch and Pitts (1943) deu início a diversos estudos sobre Redes Neurais Artificiais (RNA), uma vez que buscou entender qual era o funcionamento dos neurônios. Após este entendimento, foi criado um modelo computacional para redes neurais baseado em matemática e algoritmos denominado lógica limiar, simulando o funcionamento dos neurônios. Haykin (2001) classifica RNA como uma máquina poderosa e complexa, capaz de processar uma grande quantidade de informações em tempo mínimo e que tem como característica possuir a capacidade de simular o comportamento do cérebro ao realizar uma tarefa. Para Janidarmian et al. (2017), o RNA é um sistema de neurônios interconectados, capazes de criar modelos matemáticos para realizar o aprendizado de máquina.

Para Haykin (2001), RNA é formada por unidades básicas de processamento, chamadas de perceptrons, sendo compreendida como uma unidade de tomada de decisão com várias entradas e uma única saída. McCulloch and Pitts (1943) classifica um neurônio como uma unidade de processamento da informação que é fundamental para a operação do cérebro.

Figura 7 – Modelo de Neurônio

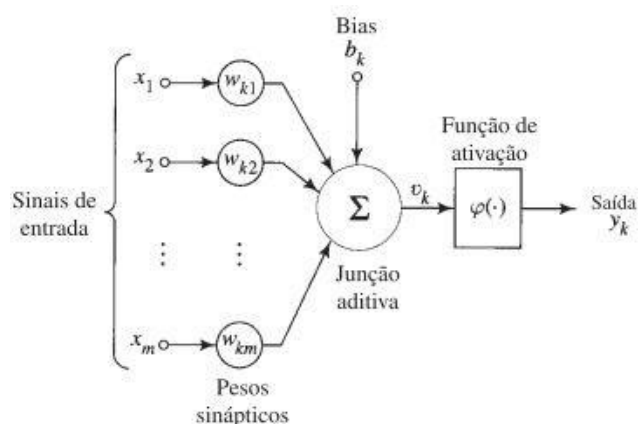


Figura 7 - Redes neurais artificiais de conexões sinapses com seus pesos. Fonte: McCulloch and Pitts (1943)

A Figura 7 apresenta um conjunto de conexões, sinapses ou elos de conexões, definido por um peso ou força própria. Especificamente, o sinal de entrada (X_j) da sinapse conectado ao neurônio k está sendo multiplicado pelo peso sináptico (W_{ji}). Após realizar o cálculo de cada neurônio, um somador é utilizado para somar os sinais de entrada e uma função de ativação é utilizada para restringir a amplitude da saída de um neurônio. Como apresentado na figura, o peso sináptico possui dois índices: o primeiro refere-se ao neurônio e o segundo, ao terminal de entrada da sinapse ao qual o peso se refere. O Bias representado por (B_b) exerce um papel importante junto a função de ativação; ele ajuda a controlar o valor no qual a função de ativação será ativada Haykin (2001).

Com os avanços da tecnologia e a evolução sobre os conceitos de RNA, diversas arquiteturas para AM foram sendo criadas. As próximas seções abordam alguns modelos de arquitetura baseado em RNA para realizar o RAH.

2.4.3.2 Feedforward Neural Network

Segundo Rosati et al. (2018), Feedforward Neural Network (FNN), também conhecida como Perceptron de Múltiplas Camadas, é uma arquitetura baseada em RNA amplamente utilizada em pesquisas científicas. Sua característica de divisão em camadas possibilita o aprendizado não linear entre entrada e vetores de saída com funções de transferência não lineares. Janidarmian et al. (2017) acrescenta que FNN é um modelo que tem como característica se alimentar e, em seguida, enviar os dados

para a próxima camada, ou seja, processar os dados na camada atual e enviar para a próxima o resultado baseado na função de ativação.

Ainda de acordo com Rosati et al. (2018), nesse modelo, para cada valor de entrada é requerido um neurônio. A camada de saída está totalmente vinculada ao número de classes que se deseja reconhecer, sendo assim o número de neurônios na camada de saída variável de acordo com o problema proposto. As camadas ocultas, cujos nós computacionais são chamados de neurônios ocultos, podem ser adicionadas entre a camada de entrada e a camada de saída, às quais o número vai depender de testar diferentes configurações. A função da camada oculta é intervir entre as camadas de entrada e saída, fazendo com que a rede possa extrair estatísticas de ordem elevada (Janidarmian et al., 2017). A Figura 8 ilustra um modelo de uma rede neural com 10 neurônios de entrada, 4 neurônios ocultos e 2 neurônios de saída, totalmente conectada; ou seja, todos os nós da camada de rede estão conectados aos das camadas subjacentes. A fórmula básica da FNN segue abaixo:

$$y = f\left(\sum_i \omega_i \cdot x_i\right)$$

Nessa fórmula matemática, y é a saída do neurônio, w_i são pesos das conexões de entrada, x_i são entradas para os neurônios e f é chamado de função de transferência, devendo ser selecionado de acordo com o problema de classificação.

Uma limitação da arquitetura MLP é assumir que todas as entradas e saídas são independentes. Quando é necessário resolver um problema de série temporal (como um sinal do sensor), é preciso incluir algumas informações temporais nos dados de entrada (Aguileta et al., 2019); Para resolver esse problema, outras arquiteturas foram projetadas, como RNN e LSTM, que serão descritas nas próximas seções.

Figura 8 – Modelo de Arquitetural FNN

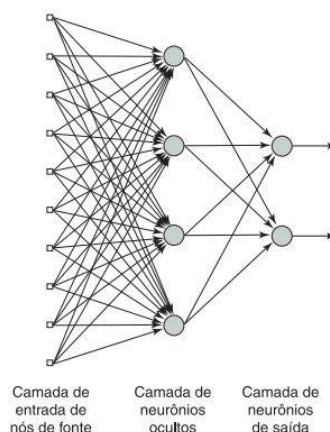


Figura 8 - Modelo arquitetural FNN com sua camada de entradas, oculta e camada de saída.

Fonte: Aguilera et al. 2019

2.4.3.3 Redes Neurais Recorrentes

Ordonez and Roggen (2016) definem Recurrent Neural Networks (RNN) como uma arquitetura baseada em RNA, projetada para resolver problema de séries temporais. A conexão entre os neurônios tem ciclo direto, ou seja, o resultado do neurônio de saída é dependente do estado da rede em dado momento passado. Isso torna-se possível por meio do *loop de feedback* conectado às suas decisões anteriores, gerando suas próprias saídas Viswambaran et al. (2019). A Figura 9 ilustra o *loop de feedback* com o primeiro neurônio (RNN $X_t H_t$) sendo realimentado pelos neurônios, em que X representa o tempo e H a saída da célula.

Figura 9 – A Estrutura Básica de Redes Neurais Recorrentes

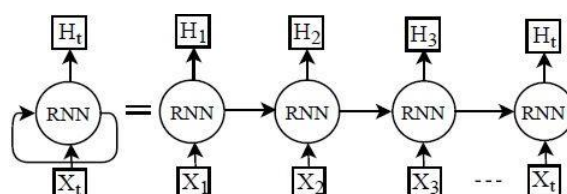


Figura 9 - Rede Neurais recorrentes com o *loop de feedback* conectados ao estado em momento passado. Fonte: Ordonez and Roggen (2016)

De acordo com Aguilera et al. (2019), devido a realimentação realizada através do *loop de feedback*, é possível armazenar as informações por meio do tempo em seu

estado oculto, dando a possibilidade de encontrar padrões em sequência de dados de longo tempo. O estado oculto recebe o valor no momento t definido por h_t , sendo esta uma função da entrada no mesmo passo do tempo x_t , e modificada por uma matriz de peso adicionada ao estado oculto no tempo anterior h_{t-1} , multiplicado por seu próprio estado oculto (Ordenez and Roggen, 2016). Uma representação de uma única unidade recorrente do modelo RNN é apresentado na Figura 10.

Figura 10 – Recurrent Neural Networks (RNN) com Duas Camadas Densas

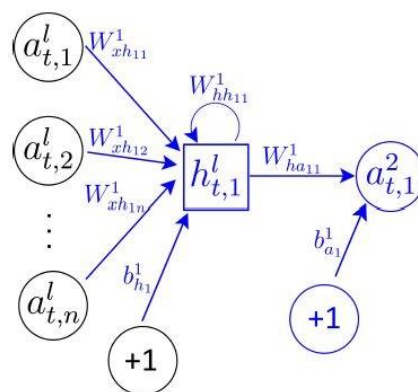


Figura 10 - Rede neurais recorrentes com o cálculo e peso. Fonte: Ordenez and Roggen, 2016

Dada uma sequência de entrada temporal $a^l = (a_1^l, \dots, a_t^l)$ de tamanho t (sendo a_t^l a unidade de ativação i na camada oculta l no tempo t), uma RNN mapeia para uma sequência de valores ocultos $h^l = (h_1^l, \dots, h_t^l)$ e gera uma sequência de ativações $a^{(l+1)} = (a_1^{(l+1)}, \dots, a_t^{(l+1)})$ por iterar a seguinte equação recursiva:

$$h_t^l = \sigma(W_{x_h}^l a_t^l + h_{t-1}^l W_{h_h}^l + b_h^l)$$

onde σ é uma função de ativação não linear, b_h^l é um vetor de bias oculto e o termo W é o peso da matriz, $W_{x_h}^l$ sendo a matriz de peso da entrada oculta e $W_{h_h}^l$ a matriz de peso oculto Ordenez and Roggen (2016). A ativação da unidade recorrente é dada pela seguinte equação:

$$a_t^{(l+1)} = h_t^l W_{h_a}^l + b_a^l$$

onde $W_{h_a}^l$ representa a matriz de peso da ativação oculta e b_a^l representa o vetor de bias.

Quando aplicado na prática para aplicar soluções de longa dependência, o algoritmo usado para a correção dos parâmetros internos não se comporta bem. Durante o treinamento, quando a derivada da função de erro com relação ao peso da rede se torna muito grande ou próximo de zero, ocorre um problema conhecido como explosão do gradiente ou *vanish gradient*. Esse problema resultará em um impacto adverso na atualização de peso pelo algoritmo de retropropagação (Long et al., 2019). Para resolver tal problema, foi criada uma arquitetura que seria uma extensão da RNN, conhecida como LSTM, que será descrita na próxima seção.

2.4.3.4 Long Short Term Memory

Com o objetivo de resolver o problema de explosão do gradiente ou *vanish gradient* (descrito na seção anterior), comumente encontrado em RNN, Hochreiter and Schmidhuber (1997) propuseram a arquitetura Long Short Term Memory (LSTM). Este modelo é uma derivação de RNN capaz de aprender e armazenar dependências de longo prazo por armazenar as informações em uma memória interna Wang et al. (2017). LSTM se difere de RNN quanto ao mecanismo de armazenamento de informações: o RNN utiliza unidades recorrentes para realizar o armazenamento, enquanto o LSTM utiliza do conceito de portões em que um mecanismo é baseado na multiplicação da entrada, definindo o comportamento de cada célula de memória individualmente (Ordonez and Roggen, 2016). A Figura 11 ilustra um modelo LSTM:

Figura 11 – Modelo de Arquitetural Recurrent Neural Networks (LSTM)

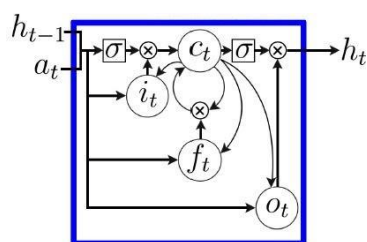


Figura 11 - Rede neurais LSTM formada por uma célula de memória auto-conectada calculando o estado através dos portões de entrada, saída e esquecimento. Fonte: Ordonez and Roggen, 2016

Conforme apresentado na Figura 11, uma unidade LSTM é formada por uma célula de memória auto-conectada e três portões: o portão de entrada (i_t), que controla o armazenamento dos dados de entrada; o portão de saída (f_t), que controla a

remoção dos estados anteriores; e o portão de esquecimento (o_t), que gera a saída do resultado (Yu et al., 2018). Esses portões são usadas para regular a atualização das informações da célula de estado. Abaixo, segue a equação para atualização do estado celular:

$$\begin{aligned} i_t &= \sigma(W_a i a_t + W_h i h_t - 1 + W_c i c_t - 1 + b_i) \\ f_t &= \sigma(W_a f a_t + W_h f h_t - 1 + W_c f c_t - 1 + b_f) \\ ct &= \sigma(f_t c_t - 1 a_t + i_t \sigma_c(W_a c a_t + W_h c h_t - 1 + b_c)) \\ o_t &= \sigma(W_a o a_t + W_h o h_t - 1 + W_c o c_t - 1 + b_o) \\ h_t &= o_t \sigma_h(c_t) \end{aligned}$$

onde, i , f , o e c são, respectivamente, os portões de entrada, de esquecimento, de saída e um vetor de ativação da célula, todos com o mesmo tamanho do vetor h , definindo o valor oculto. O termo σ representa uma função não linear. O termo a é a entrada para a camada da célula de memória no momento t . $W_a i$, $W_h i$, $W_c i$, $W_a f$, $W_h f$, $W_c f$, $W_a c$, $W_h c$, $W_a o$, $W_h o$ e $W_c o$ são matrizes de peso, com subscritos representando relações de-para (sendo $W_a i$ a matriz de entrada, etc.). b_i , b_f , b_c e b_o são vetores de viés (Aguileta et al., 2019).

2.4.3.5 Gated Recurrent Unit

Proposta por Chung et al. (2014), a Gated Recurrent Unit GRU é uma extensão de RNN, capaz de armazenar os dados e os pesos em memória, tendo sua arquitetura similar ao LSTM, porém com implementação e poder de computação mais simples. Sua arquitetura é formada por dois portões, um portão de atualização (z) e um de esquecimento (r); ambos repassam informações por meio de muitas janelas de dados, buscando atingir uma melhor predição ou classificação (Lei et al., 2018). A Figura 12 ilustra a arquitetura GRU.

2.4.3.6 Convolutional Neural Network

Convolutional Neural Network (CNN) é um tipo de RNA capaz de executar operações convolucionais (multiplicação de matrizes) em filtros (matriz) para extrair um padrão de características sobre um conjunto de dados sem a necessidade de um especialista (Aguileta et al., 2019; Gil-Martín et al., 2019). Este modelo é intimamente relacionado ao FNN, mas se diferencia por adicionar em sua arquitetura o conceito de camada de convolução e camada de *pooling* (Yu et al., 2018; Aguileta et al., 2019).

Camadas convolucionais são um importante componente de CNN. Com a utilização de vários filtros (ou *kernel*) convolucionais, é possível extrair representações dos dados de entrada (Input) sem a necessidade de manipulá-los ou pré-processá-los (Aguileta et al., 2019). Em CNN, a ativação de uma unidade realiza-se por meio do resultado da convolução do kernel com os dados de entrada. Ao calcular a ativação de uma unidade em diferentes regiões dos dados de entrada, é possível detectar padrões de características capturadas pelos *kernels*, independentemente de onde o padrão ocorre. O resultado da etapa de convolução é um mapa de características que conectados pelos neurônios de camadas posteriores Ordonez and Roggen (2016). O valor do mapa de características será por maps na camada l, $x_j^{(l+1)}$ calcula-se por:

$$x_j^{(l+1)} = \sigma \left(\sum_{maps} x_i^{(l)} \otimes k_{ij}^{(l)} + b_j^l \right)$$

onde maps é o total de características do mapa na camada l-th, e $b_j^{(l)}$ é um vetor de bias. O σ é uma função de ativação. A Figura 13 ilustra o resultado de operação do kernel.

Figura 13 – Resultado da aplicação da camada convolucional (CNN)

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

Fonte: Aguilera et al. (2019)

A camada de *pooling* é uma camada incluída pela arquitetura CNN para diminuir a dimensão espacial da saída da camada convolucional sem alterar a profundidade. Essa camada traz vantagens em se tratando de operações computacionais por diminuir o número de operações, evitando assim a adaptação do modelo durante treino de dados. A camada de *pooling* contém *min*, *max* e média operações (Danaei-mehr and Polat, 2019).

As equações para calcular a altura (H2) e o tamanho (W2) da saída dessa camada se dão por:

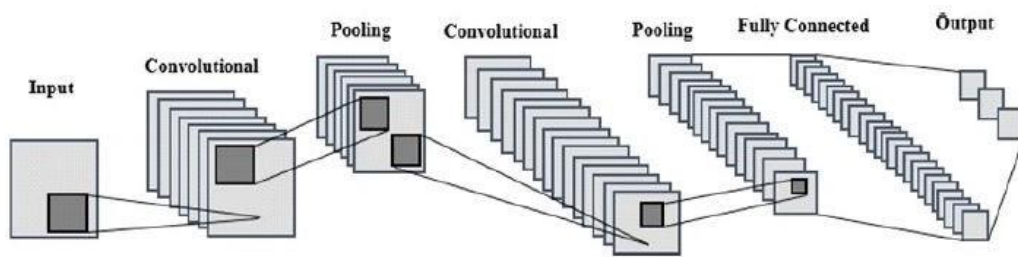
$$W2 = \left(\frac{W_i - F}{S} \right) + 1$$

$$H2 = \left(\frac{H_i - F}{S} \right) + 1$$

onde W2 representa o tamanho, H2 se refere à altura e F e S são, respectivamente, o tamanho do *kernel* e o tamanho da passada de mudança (Danaei-mehr and Polat, 2019).

Uma terceira camada utilizada pela arquitetura CNN é a camada totalmente conectada. Essa camada é responsável não apenas por conectar todos os neurônios da camada anterior, como também por realizar o cálculo de pontuação sobre as classes de um conjunto de dados (Danaei-mehr and Polat, 2019). Funções não lineares também se aplicam nessa camada, geralmente a função *softmax* é para calcular a distribuição dos rótulos das classes. A Figura 14 ilustra as 3 camadas apresentadas nesta seção, nela é possível visualizar cada camada.

Figura 14 – Modelo de Arquitetura Recurrent Neural Networks (LSTM)



Fonte: Danaei-mehr and Polat (2019)

3 TRABALHOS RELACIONADOS

Nos últimos anos, houve um aumento significativo no número de pesquisas relacionadas ao reconhecimento de atividades humanas (RAH). Diversos pesquisadores abordaram diferentes aspectos com o objetivo de melhorar o desempenho ao realizar o RAH, extração de dados de imagens, sensores, estratégia de aquisição de dados e técnicas de AM para classificação das atividades. Todas essas estratégias têm impactos cruciais no desempenho dos sistemas RAH. Nesta seção, a partir de revisão bibliográfica realizada neste projeto de pesquisa, apresenta-se as abordagens usadas por diversos pesquisadores.

3.1 Técnicas de Machine Learning

O trabalho proposto por Long et al. (2019) e Ordonez and Roggen (2016) criou uma estrutura de aprendizado para reconhecimento de atividades usando sensores. A arquitetura foi formada por duas técnicas de máquina de aprendizado, Convolutional Neural Network (CNN) e Long Short Term Memory (LSTM). Para o treinamento do modelo, foi utilizado o *dataset* OPPORTUNITY, proposto por Roggen et al. (2010). No presente trabalho, seguimos o modelo de LSTM proposto pelos autores, diferenciando apenas em arquitetura do modelo, números de camadas, neurônios e algoritmos de treinamento e aplicando-o sobre o conjunto de dados UniverCemig.

Os pesquisadores Gil-Martín et al. (2019) realizaram a implementação de Convolutional Neural Network (CNN) para realizar o RAH. Os dados das atividades executadas foram extraídos a partir de um acelerômetro introduzido em um *smartphone* utilizado por 24 pessoas, capturando 6 atividades. Em uma *baseline* utilizando SVM, a técnica implementada melhorou o resultado em aproximadamente 9%. Por SVM ter apresentado um excelente resultado, servindo até como *baseline* para futuros trabalhos, no presente projeto seguimos essa prática, diferenciando a abordagem de coleta de dados e a implementação do SVM conforme pode ser visto na análise de resultados.

A abordagem proposta por Gumaei et al. (2019), como uma alternativa ao LSTM, utiliza uma estrutura baseada em multisensores para o reconhecimento da atividade humana, usando um modelo híbrido de aprendizado profundo que combina as unidades recorrentes simples (URS) com as unidades recorrentes (GRU) baseado em

redes neurais. Neste trabalho utilizamos a técnica proposta GRU, aplicando-a no conjunto de dados UniverCemig (coletados durante o desenvolvimento desse trabalho) com o objetivo de comparar os resultados com a técnica LSTM.

Tasnim et al. (2020) propuseram uma abordagem utilizando algoritmos de aprendizado profundo (LSTM e CNN) com os conjuntos de dados UTD-MHAD e MSR-Action3D, alcançando um desempenho de aproximadamente 96% em ambos os conjuntos de dados. Esta projeto serviu como referência por ser aplicado em conjunto de dados com coordenadas de referência do corpo humano extraídos de imagens, utilizando técnicas de extração de dados em 3D, gerando as coordenadas x, y e z de um movimento executado. Porém, no presente trabalho propusemos a implementação de extração de imagem em 2D extraindo as coordenadas x e y das imagens obtidas a partir de uma filmagem, conforme pode ser visto na seção 5, coleta de dados.

A pesquisa realizada por Aguilera et al. (2019) direcionou o presente trabalho por apresentar os resultados de várias pesquisas e variadas técnicas de AM como SVM, KNN, CNN e LSTM, implementadas para realizar o RAH pela aplicação de técnicas de extração, fusão e segmentação de dados de sensores e imagens.

O presente projeto de pesquisa baseia-se em trabalhos anteriores como Long et al. (2019), Ordonez and Roggen (2016) e Tasnim et al. (2020), utilizando dados extraídos de imagens de com o objetivo de classificar as atividades executadas durante o treinamento do aluno no curso de eletricista. Serão criadas duas abordagens para análise e comparação dos resultados: A primeira será a implementação de SVM e KNN, que será utilizada como *baseline*; a segunda será a implementação do LSTM e do GRU, técnicas de aprendizado de máquina que têm sido amplamente aplicadas nos dias atuais.

4 METODOLOGIA

A metodologia tem como objetivo orientar o pesquisador, direcionando os estudos, garantindo que os objetivos propostos sejam cumpridos. A ciência é caracterizada pelos métodos científicos que devem ser aplicados, explicados, evidenciando a abordagem do estudo Córdova and Silveira (2008). Este capítulo descreve a metodologia de pesquisa utilizada neste estudo para alcançar os objetivos propostos, a caracterização da pesquisa, os procedimentos para a coleta de dados e a análise de resultados.

4.1 Caracterização da Pesquisa

A pesquisa deste trabalho caracteriza-se como aplicada. De acordo com Córdova and Silveira (2008), a pesquisa aplicada objetiva investigar, gerar conhecimentos para aplicação prática e comprovar ou rejeitar hipóteses. Executaremos esta pesquisa com o propósito de identificar e investigar técnicas de aprendizado de máquinas aplicadas no problema de reconhecimento de atividades humanas.

A natureza de pesquisa deste trabalho foi de caráter quantitativa. A abordagem quantitativa tem por objetivo garantir uma precisão sobre os dados analisados, assegurando um aumento da margem de confiança sobre estes para uma tomada de decisão com a utilização de técnicas estatísticas Baptista and Cunha (2007). Neste trabalho, a estatística será utilizada durante a etapa de análise e resultados.

Este trabalho tem como objetivo a pesquisa exploratória. Segundo Córdova and Silveira (2008) a pesquisa exploratória caracteriza-se por desenvolver, esclarecer e modificar conceitos e ideias, tendo em vista a formulação de problemas ou hipóteses pesquisáveis para estudos posteriores. O caráter de pesquisa exploratório se justifica devido à natureza deste trabalho, que tem como objetivo a implementação de técnicas de aprendizado de máquina capazes de reconhecer as atividades executadas por um ser humano.

4.2 Paradigma da Pesquisa

O paradigma a ser aplicado no trabalho será o positivista, considerando verdades objetivas, independente da percepção humana. Saccol (2010) considera

que a realidade é composta por estruturas palpáveis, tangíveis e relativamente estáveis. Meramente fenômenos mensuráveis estão aptos a construir conhecimento verdadeiramente válido. De acordo com o positivismo, cada conceito e cada idéia necessitam ser operacionalizados, isto é, definidos claramente e transformados em uma ou mais variáveis que possam ser observadas e medidas de forma objetiva.

4.3 Estratégias da Pesquisa

Para o desenvolvimento desta pesquisa, utilizou-se a estratégia de pesquisa bibliográfica. Segundo Gil (2009), a pesquisa bibliográfica é um dos passos significativos ao iniciar o processo de pesquisa para um trabalho científico, permitindo ao pesquisador entender o assunto da pesquisa por intermédio de materiais anteriormente publicados. Para este trabalho, a pesquisa bibliográfica teve como objetivo elementar o levantamento de referências teóricas sobre técnicas de máquina de aprendizado para o RAH.

Entende-se por pesquisa bibliográfica o estudo sobre as teorias vitais que norteiam o trabalho científico. Essa revisão pode ser definida como levantamento bibliográfico. Nesta etapa, é realizada a pesquisa de referências teóricas analisadas e publicadas por via de escritos e eletrônicos, como livros, revistas, artigos científicos, publicações em periódicos, monografias, dissertações e *web sites* com o objetivo de colocar o pesquisador em contato direto com o material escrito sobre o assunto da pesquisa, facilitando a identificação e a seleção de métodos e técnicas a serem utilizados, oferecendo subsídios para a redação da introdução e da discussão do trabalho científico (Prodanov and C.Freitas, 2013). A tabela 1 apresenta os passos essenciais para alcançar cada objetivo do presente trabalho.

Tabela 1 – Passos para alcançar os objetivos deste trabalho

Objetivo	Definições
Criar um conjunto de dados (<i>dataset</i>) com as informações obtidas de imagens	<ul style="list-style-type: none"> • Identificar as atividades executadas • Identificar os participantes • Filmar o treinamento

	<ul style="list-style-type: none"> • Segmentar o vídeo e gerar imagens • Extração de pontos de referência do corpo humano a partir das imagens • Anotar as atividades executadas • Gerar artefato com dados das atividades
Implementar técnicas de aprendizado de máquina e aprendizado de máquina profundo no reconhecimento de atividades humanas	Aplicar Técnicas de Aprendizado de Máquina (KNN e SVM) e aprendizado de máquina profundo sobre os dados coletados (LSTM e GRU)
Analisar os resultados dos técnicas implementadas	Analisar os resultados das técnicas implementadas através de métricas, como: Acurária, Precisão e Revocação

Tabela 1 – Passos essenciais para que todos os objetivos definidos no presente trabalho possam ser alcançados. Fonte: Elaborado pelo autor.

4.4 Proposta de Modelo

Com o propósito de entender como se dará o fluxo desta pesquisa, a Figura 15 ilustra os passos que serão executados no trabalho, cujos elementos serão explicados individualmente nas seções posteriores.

Figura 15 – Proposta de Modelo de Execução do Projeto



Figura 15 - Passos para a realização do projeto de pesquisa. Fonte: Elaborado pelo autor.

4.4.1 Identificar Atividades executadas

A UniverCermig define um conjunto de atividades essenciais à formação do eletricitista. No entanto, para o presente trabalho, iremos coletar dados da atividade de montagem de estrutura B1. Nessa atividade, o aluno deve seguir todas os procedimentos e normas que foram desenvolvidos na parte teórica do curso. A montagem deste tipo de estrutura envolve algumas tarefas: assim, segmentamos tal atividade conforme listagem:

- Fixação de Bracete
- Fixação de Bastão
- Fixação de Pinos

4.4.2 Coleta de dados

Esta etapa refere-se ao processo de extração dos dados gerados. O fluxo de coleta de dados deve ser estabelecido de modo que possam ser compreendidos todos os passos e variáveis valiosas para o modelo que será utilizado pelas técnicas de ML para classificação da atividade. Realizar a coleta de dados é um dos objetivos específicos deste trabalho 1. A Figura 16 ilustra o fluxo de coleta de dados e as seções posteriores descreverão cada etapa:

Figura 16 – Fluxo de coleta dos dados

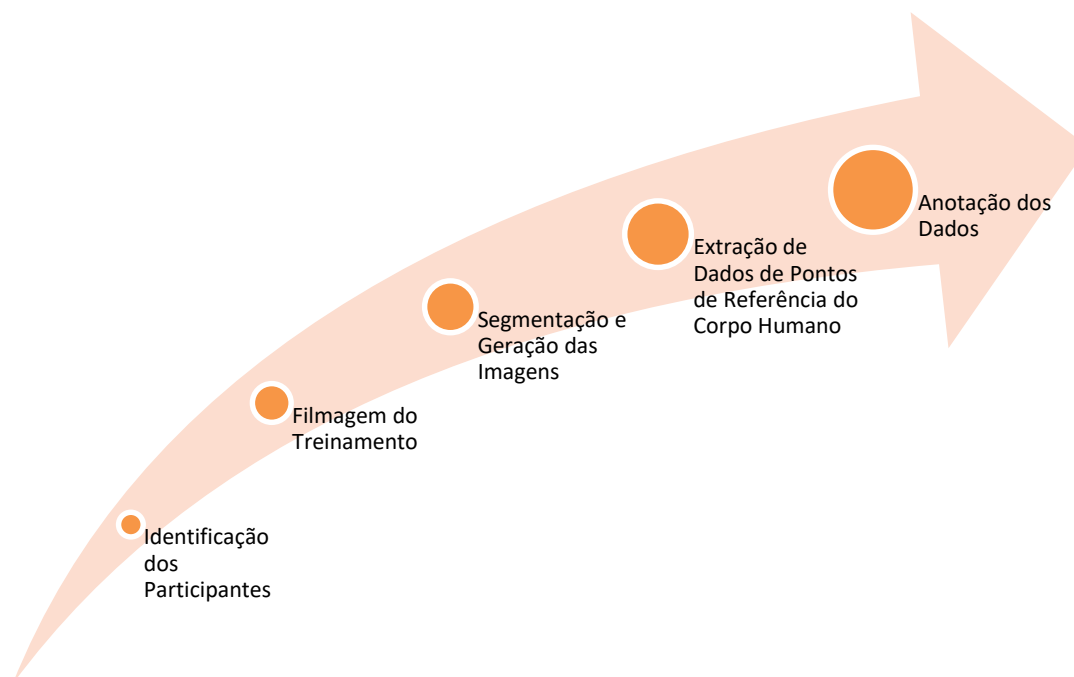


Figura 16 - Descrição dos passos para a realização da coleta de dados. Fonte: Elaborado pelo autor.

4.4.2.1 Identificação dos Participantes

Nesta etapa, serão definidos todos os participantes que estarão envolvidos no processo de treinamento. Durante ela, é necessário identificar alguns dados que serão de importância para a pesquisa e essenciais durante o processo, sendo eles:

- Quantidade de participantes;
- Atividade executada;
- Aluno ou Professor.

4.4.2.2 Filmagem do Treinamento

Devido à natureza da solução proposta pelo presente trabalho, esta etapa tem como objetivo realizar a filmagem do processo de montagem da estrutura B1 no treinamento realizado pelos alunos da UniverCemig, para que posteriormente os dados possam ser extraídos com as técnicas que serão explicadas nas seções seguintes.

4.4.2.3 Segmentação do Vídeo e Geração de Imagens

Após a realização de toda a filmagem referente à atividade de montagem de estrutura B1 (4.4.2.2), foi necessário segmentar o vídeo por atividade executada por cada aluno. Feita a segmentação, será executada uma técnica para extração de imagens a partir dos vídeos gerados.

4.4.2.4 Extração de Pontos de Referência do Corpo Humano

Pontos de referência do corpo humano (2.3) fornecem informações espaciais de cada membro do corpo humano em uma imagem. Nesta etapa, iremos executar técnicas de AM para extração dos pontos de junção a partir dos imagens geradas.

4.4.2.5 Anotação da Atividade

Esta etapa consiste em identificar a atividade executada e realizar a anotação referente a ela mediante uma descrição em formato numérico, para que este dado seja usado pelas técnicas de AM.

4.4.2.6 Artefato Gerado

Executados todos os passos conforme apresetado nas seções anteriores no fluxo de coleta de dados, a saída deste processo será um arquivo no formato **.csv** com a anotação das atividades que foram executadas. Cada linha desse arquivo deverá conter 34 registros separados por vírgula.

4.4.3 Aplicar Técnicas de Aprendizado de Máquina

Os resultados da pesquisa bibliográfica possibilitaram encontrar trabalhos que visavam resolver o problema de RAH utilizando técnicas de AM e AMP. O capítulo 2 apresentou os principais tópicos relativos ao RAH. Conforme apresentado na seção 2.2 o reconhecimento de atividades humanas é um problema de classificação que envolve dados de séries temporais e, com o objetivo de resolver esse problema, apresentamos as seções 2.4 e 2.4.3.

Este trabalho objetiva implementar técnicas de máquina de aprendizado respondendo ao item 2 dos objetivos específicos. Assim, a implementação divide-se em duas abordagens. A primeira abordagem será uma implementação das técnicas

de AM SVM e KNN, como visto na seção 2.4. A segunda abordagem será a implementação dos modelos de AMP, GRU e LSTM, apresentados na seção 2.4.3.

4.4.4 Análise dos Resultados

Esta etapa tem como objetivo verificar, através de métricas, os resultados das técnicas de AM implementadas, pois trabalhos que envolvem comparações de qualquer natureza requerem a definição acerca da ótica na qual os dados devem ser analisados (Córdova and Silveira, 2008). É imprescindível que o reconhecimento de atividades humanas no contexto do treinamento do eletricista não rotulem atividades erroneamente, isso poderia trazer prejuízos no processo de certificação do aprendiz do aluno.

De acordo com Gumaei et al. (2019), é factível utilizar métricas de acurácia, precisão e revocação para avaliação dos modelos de AM que correspondem às ausências de resultados falso-positivos e falso-negativos, respectivamente.

Para tal, é necessário calcular as medidas Precisão e Revocação para cada classe $(C_i) \in C$. Para as equações seguintes, considere que:

- VP_{c_i} (Verdadeiros positivos) representa o número de registros de teste que foram corretamente atribuídos à classe c_i .
- FP_{c_i} (Falso positivos) representa o número de registros de teste que foram corretamente atribuídos à classe $(C_j)(c_j \neq c_i)$, mas atribuídos à classe c_i .
- FN_{c_i} (Falsos negativos) representa o número de documentos de teste da classes c_i rotulados com uma classe $(C_j)(c_j \neq c_i)$.

4.4.4.1 Acurácia

Indica uma performance integral do modelo e, dentre todas as classificações, quantas o modelo classificou corretamente.

$$Acurácia(c_i) = \frac{VP_{c_i} + VNC_i}{VP_{c_i} + VNC_i + FN_{c_i} + FP_{c_i}}$$

4.4.4.2 Precisão

Pode ser vista como uma medida de exatidão, pois denota a ausência de positivos fictícios, retornando a porcentagem de dados corretamente rotulados pelo classificador.

$$Precisão(c_i) \frac{VPC_i}{VPC_i + FPC_i}$$

4.4.4.3 Revocação

Dentre todas as situações de classe Positivo como valor esperado, quantas estão corretas.

$$Revocação(c_i) \frac{VPC_i}{VPC_i + FNC_i}$$

5 COLETA DE DADOS

Este capítulo apresenta como foi realizado o processo para a coleta de dados. Tal processo envolve várias etapas que serão descritas individualmente em cada seção. Por fim, apresentaremos o resultado do conjunto de dados gerado.

5.1 Abordagem

Devido à inexistência de um conjunto de dados que contemple as atividades executadas por um profissional de elétrica no seguimento de energia elétrica para RAH, foi criado um conjunto de dados com características específicas das atividades de um eletricista, coletados durante o curso de capacitação oferecido pela UniverCemig. O conjunto de dados extraído pode ser utilizado como entrada para a criação de modelos de AM para classificar as atividades executadas pelos alunos durante o treinamento, uma vez que o ambiente de campo do curso possui características que deixam o aluno o mais próximo possível do cenário real vivenciado em sua profissão.

5.1.1 Filmagem do Treinamento

Para iniciarmos o processo de coleta de dados, o primeiro passo realizado foi a filmagem do aluno em treinamento. Para esta etapa, foram selecionados de forma aleatória pelo coordenador do curso 4 alunos que participaram de todas as etapas do curso anteriores ao treinamento de campo, estando, assim, aptos a realizar essa fase do treinamento. Estes alunos foram orientados a executar a atividade de montagem de estrutura B1 de forma independente, seguindo os padrões propostos durante todo o curso. Para esta etapa, foram utilizados uma webcam e um notebook Predator Helios 300, placa Geforce GTX 1060, 16 GB de memória, gerando um vídeo de 01:03:55 horas de filmagem.

5.1.2 Segmentação do Vídeo, Geração de Imagens e Filtragem

Após realizar a filmagem do processo de montagem de estrutura B1 (5.1.1),

extraiu-se imagens do vídeo. Nesta etapa, o vídeo foi segmentado por participantes, gerando 4 vídeos contendo todas atividades executadas por cada indivíduo. Cada vídeo gerado foi segmentado em 44 *frames* por segundo, ou seja, para cada segundo do vídeo foram geradas 44 imagens, totalizando 127.856 imagens.

Neste mesmo processo, foram eliminadas todas as imagens em que não havia nenhuma pessoa no ambiente de montagem de estrutura; isso ocorreu devido ao rodízio de alunos durante a filmagem.

Para geração de imagens, foi utilizada a biblioteca OpenCV versão 4.5.1, linguagem de programação Python versão 3.7 e IDE para desenvolvimento Pycharm. O processamento de todo o vídeo durou cerca de 9 horas em um *notebook* Predator Helios 300, placa Geforce GTX 1060, 16 GB de memória.

5.1.3 Pontos de Referência do Corpo Humano

Conforme explicado na seção 2.3, pontos de junção do esqueleto fornecem informações sobre a posição de um membro do corpo em um espaço. Com o objetivo de extrair essas informações, o presente trabalho utilizou o *framework* TensorFlow PoseNet, que usa o modelo CNN de AM para analisar a pose de uma pessoa a partir de uma imagem, estimando as localizações espaciais das principais articulações do corpo humano.

Cada articulação é mapeada por um ID e retorna as coordenadas x e y com uma pontuação de confiança entre 0,0 e 1,0. Com o objetivo de validar os dados retornados foi configurado o parâmetro *min-pose-score* com valor de 70%. A pontuação de confiança indica a probabilidade de que um ponto-chave exista nessa posição.

Para estimar a pose, foi utilizado o modelo MobileNet. Cada imagem, quando submetida ao *posenet*, gera 17 características e, para cada uma, seu valor de coordenada x e y como mostra a Figura 18. O processamento de todas as imagens durou cerca de 3 horas em um *notebook* Predator Helios 300, placa Geforce GTX 1060, 16 GB de memória.

Figura 17 – Pontos de Referência do Corpo Humano

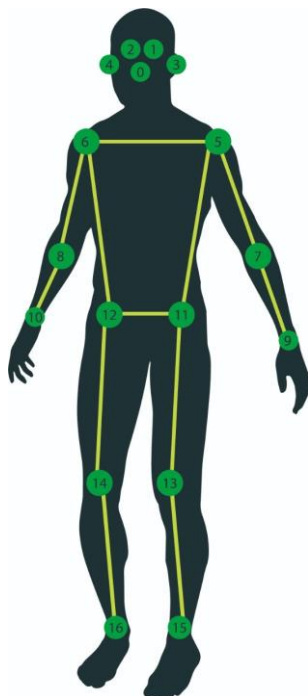


Figura 17 - Cada ponto é marcado por um membro do corpo humano que retorna as coordenadas X e Y no espaço dimensional. Fonte: Pardos et al. (2020)

Segundo Pardos et al. (2020), PoseNet é uma combinação de visão computacional e AMP que utiliza um modelo pré-treinado para estimar a pose de uma pessoa em uma imagem ou vídeo. Para o presente trabalho, o PoseNet foi treinado com o modelo MobileNet,5.1. que utiliza uma arquitetura CNN para estimar as localizações espaciais das principais articulações do corpo. O processamento no MobileNet é feito em duas etapas: a primeira é a fatorar em uma camada de convolução para filtragem e uma camada de convolução 1 x 1 combinado.

Quando uma imagem é processada, o resultado é um mapa de calor junto com vetor de deslocamento. Estes podem ser decodificados para encontrar áreas de alta confiança na imagem, que correspondem aos pontos-chave de pose. O vetor de deslocamento é um tensor 3D com tamanho Largura x Altura x ((pontos chaves)17 * 2 (coordenadas X e Y)), usado para localizar pontos na imagem fornecida .

A Figura 18 apresenta o mapa de calor de detecção do ponto de referência do pulso direito.

Figura 18 – Mapa de calor representado o ponto-chave pulso direito

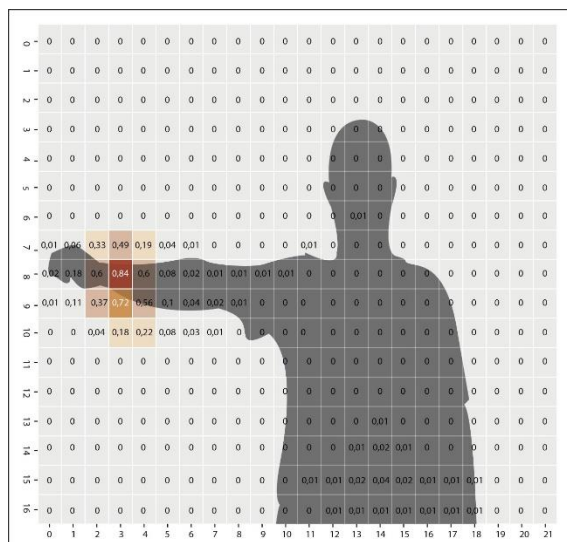


Figura 18 - Mapa de calor representado as referências X e Y do pulso direito. Fonte: Elaborado pelo autor.

5.1.4 Processo Manual de Anotação dos dados

Para o processo de anotação de dados foi adotada a técnica de anotação supervisionada realizada com base no vídeo do item 5.1.2. Cada vídeo gerado foi segmentado em atividades que cada aluno executou, possibilitando anotar os dados que representassem cada atividade executada pelos indivíduos durante o treinamento.

5.1.5 Conjunto de dados

Os dados capturados foram separados em arquivos com extensão .csv contendo os dados referente às atividades executadas por cada um dos 4 alunos. Foram gerados 4 arquivos com 35 colunas, representando os pontos de junção do esqueleto humano e a atividade relacionada com um total de 98.588 instâncias. Estas representavam 3 atividades: (1) Fixação do bracelete ao poste, (2) Fixação de bastão e (3) Fixação de pinos. O conjunto de dados criado no presente trabalho ficará disponível na seguinte URL <https://github.com/rubenssantoTI/mestrado> para que possa ser utilizada em pesquisa com o objetivo de realizar o RAH, servindo, por vezes, como baseline para validação e testes de técnicas propostas. A Figura 19 representa

os dados extraídos por atividade.

Figura 19 – Gráfico com resultados por atividade

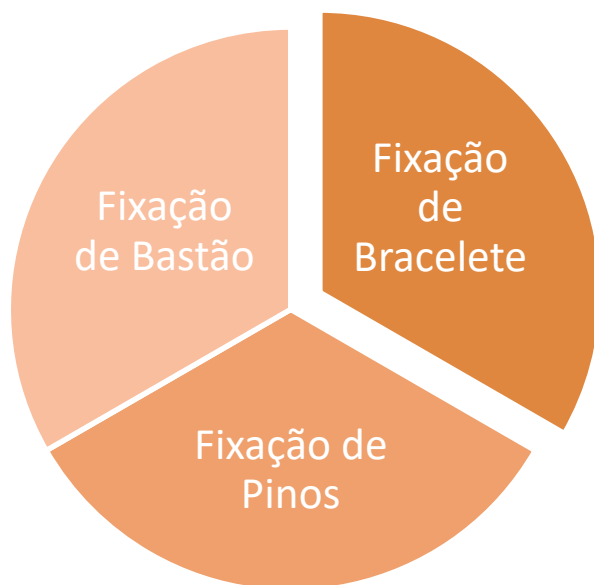


Figura 19 - Gráfico de Atividades (1) Fixação do bracelete ao poste (2) Fixação de bastão e (3) Fixação de pino. Fonte: Elaborado pelo autor.

6 RESULTADOS

O experimento do presente trabalho foi executado em um notebook Predator Helios 300, placa GTX 1060, 16 GB de memória. Para validação e implementação, utilizamos as bibliotecas *scikit-learn*, *pandas*, *numpy*, *keras*, *tensorflow* e *matplotlib*, linguagem de programação Python 3.7 e IDE de desenvolvimento Pycharm. Nas seções seguintes, iremos detalhar as configurações das técnicas utilizada e os resultados, que foram divididos em duas seções: Aprendizado de Máquina e Aprendizado de Máquina Profundo.

6.1 Experimento Aprendizado de Máquina

Esta seção apresenta os resultados e configurações provenientes das implementações de duas técnicas de AM, o KNN 2.4.2 e SVM 2.4.1, atendendo ao objetivo 2 do presente trabalho. As técnicas foram treinadas com dados coletados conforme apresentado no item 5.1.5, sendo separados em teste e treino, 25% e 75% respectivamente. Os melhores hiperparâmetros encontrados foram alcançados através de testes manuais. Escolhemos uma janela de 44 amostras por segundo, com o tempo variando entre 3 e 60 segundos de dados para extração de características que podem capturar ciclos em atividades de Fixação de bracelete, Fixação de pinos e Fixação de bastão (5.1.5).

Para reduzir a perda de informações nas bordas das janelas, utilizamos uma sobreposição de 50%. Segundo Su et al. (2014), é possível encontrar dados relevantes através de cálculos que agrupem-nos, retornando informações que possam ser utilizadas pelas técnicas de AM. Assim, para cada eixo, calculamos a raiz quadrada, o desvio padrão, o máximo e o mínimo (Su et al., 2014; Nguyen et al., 2015).

6.1.1 Hiperparâmetros K-Nearest Neighbors

Para que se chegasse ao melhor valor dos parâmetros, diversos testes manuais foram feitos. Neles, o valor de k variou entre 3 e 20. Os parâmetros para a execução das técnicas KNN são apresentados na tabela 2.

Tabela 2 – Hiperparâmetros KNN

Parâmetros da KNN	
Parâmetro	Valor
neighbors	8
Weights	uniform
metric	minkowski
p	1

Tabela 2 – Hiperparâmetros configurados para a técnica KNN. Fonte: Elaborado pelo autor.

6.1.2 Hiperparâmetros Support Vector Machine

Os testes para chegar ao melhor resultado com SVM foram realizados de forma manual, utilizando 4 funções para testar o *kernel*: radial, linear, polynomial e sigmoide. Para a função C, os testes foram variados entre 1 e 10. Os parâmetros que apresentaram melhor resultados são apresentados na tabela 3.

Tabela 3 – Hiperparâmetros de Configuração da Técnicas SVM

Parâmetros da KNN	
Parâmetro	Valor
Kernel	Polinomial
c	5
gamma	1

Tabela 3 - Hiperparâmetros configurados para a técnica SVM. Fonte: Elaborado pelo autor.

6.1.3 Análise dos Resultados de Aprendizado de Máquina

No presente trabalho, apresentamos os resultados das técnicas KNN e SVM. Na Tabela 4, exibimos os resultados do experimento para nosso modelo projetado em termos de precisão, revocação e acurácia para o conjunto de dados UniverCemig. Alcançamos precisões altas de aproximadamente 97% com o SVM e 95% com KNN. Os resultados apresentados tiveram como item fundamental a janela de tempo (2.2.2). Os testes foram executados com janelas de tempo entre 3 e 60 segundos.

Tabela 4 – Resultados das Técnicas de SVM e KNN

Janela em Segundos(S)	Algoritmo	Acurácia	Revocação	Precisão
3(S)	KNN	0.9175	0.9176	0.9178
	SVM	0.9657	0.9657	0.9658
5(S)	KNN	0.9020	0.9026	0.9015
	SVM	0.9328	0.9328	0.9329
10(S)	KNN	0.8750	0.8750	0.8895
	SVM	0.9459	0.9454	0.9472
20(S)	KNN	0.8571	0.8671	0.8675
	SVM	0.8918	0.8917	0.8975
30(S)	KNN	0.8378	0.8378	0.8412
	SVM	0.9183	0.9183	0.9339
60(S)	KNN	0.8887	0.8888	0.9206
	SVM	0.9583	0.9583	0.9642

Tabela 4 - Resultados das técnicas KNN e SVM de acordo com cada janela de tempo. definida.
 Fonte: Elaborado pelo autor.

A técnica SVM apresentou melhores resultados quando comparada com KNN mesmo quando a janela de tempo aumenta, o que pode acarretar um menor desempenho das técnicas, como pode ser observado na Figura 20. De acordo com Aguilera et al. (2019), a perda de desempenho ocorre porque uma janela de tempo maior pode capturar mais informações, o que pode afetar significativamente o desempenho do classificador. A técnica SVM não foi afetado pela janela de tempo; porém, o KNN apresentou uma leve queda de desempenho quando a janela aumenta entre 10(s) e 60(s).

Aguilera et al. (2019) propôs em seu trabalho uma revisão sobre os métodos propostos na literatura para o reconhecimento de atividade humana com dados de sensores, sendo avaliados diversos trabalhos, métodos e, posteriormente comparados. Apesar da abordagem diferente na coleta de dados, o trabalho proposto serviu com um guia no que tange as possíveis formas de implementação, técnicas e os resultados obtidos sobre cada uma. Entre as técnicas de AM, SVM foi a mais utilizada para realizar o reconhecimento de atividades humanas, implementada em 27 pesquisas; KNN apareceu em 12 pesquisas com resultados publicados. As

implementações foram realizadas sobre diversos conjunto de dados, entre eles: WISDM (Kwapisz et Al 2011) e Daphnet (Bachlin et Al 2011). Os resultados apresentados mostram que SVM teve sua acurácia entre 81% e 98%. A técnica KNN também apresentou ótimos resultados, variando entre 80% e 97%. Quando analisamos o presente trabalho, verificamos que os resultados estão dentro do esperado de acordo com trabalhos apresentados por Aguilera et al. (2019). Vale observar que as implementações das técnicas de AM e AMP foram com dados do conjunto de dados UniverCemig, criado durante a fase de coleta de dados, por não encontrarmos um conjunto de dados que contemplasse as atividades de um electricista em treinamento.

Figura 20 – Gráfico de acuracia de KNN e SVM por janela de tempo

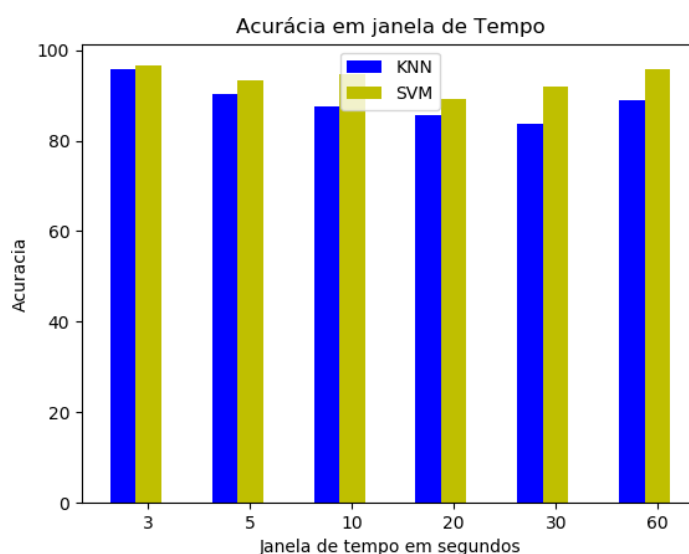


Figura 20 - Gráfico apresentado as variações das técnicas SVM e KNN ao longo da janela de tempo. Fonte: Elaborado pelo autor.

6.2 Experimento Aprendizado de Máquina Profundo

Nesta seção, iremos detalhar as configurações e resultados de duas técnicas de AMP para realizar o RAH, atendendo ao objetivo 2 do presente trabalho. Todas as técnicas citadas nas próximas seções foram treinadas através do conjunto de dados UniverCemig, conforme apresentado no item 5.1.5. Os arquivos gerados pelo conjunto de dados foram separados em treino e teste, sendo 3 arquivos para treino e 1 para teste, com uma margem de 31% e 69% para teste e treino respectivamente sem

sobreposição (2.2.2). Para iniciar o experimento das técnicas propostas, primeiro foi definido o estilo arquitetural de cada uma. Após a definição, iniciamos a busca pelo melhor tamanho do *batch* e números de épocas que iriam apresentar a melhor acurácia. Iniciamos com o número de época 1 e fomos aumentando gradualmente, até que obtivéssemos o melhor resultado. Nas seções abaixo, iremos detalhar o estilo arquitetural proposto por cada modelo, seus resultados, como foram realizados os treinamentos e as configurações utilizadas.

6.2.1 Gated Recurrent Unit

Após a definição do estilo arquitetural apresentado na tabela 4, iniciamos a execução do experimento. Durante os treinos, alternamos os valores janela de tempo entre 1 e 6 segundos. O número de épocas foi definido em 500, com tamanho de *batch* em 300. Definimos também o parâmetro paciente com o valor 30; isso significa que, caso o modelo pare de apreender, a execução do treino irá parar imediatamente após 30 tentativas. A Figura 21 apresenta o resultado do treino quando a janela de tempo foi configurada em 6 segundos. Nela, é possível observar que o modelo começou com perda e validação altas e que estas foram diminuindo através das épocas de treinamento, tendo alcançado seu melhor resultado de validação na época 225 e o melhor resultado de treino na época 200, ambos convergindo na mesma linha na época 190.

Figura 21 – Modelo de Treino Gated Recurrent Unit

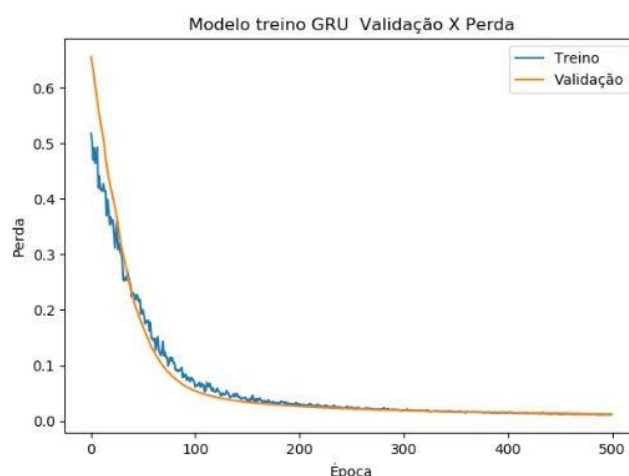


Figura 21 - Treinamento do modelo GRU apresentando a variação da validação e treino ao longo das épocas. Fonte: Elaborado pelo autor.

Tabela 5 – Configuração Modelo GRU e Hiperparâmetros

Hiperparâmetros GRU			
Camada	Função de Ativação	Número de Neurônios	Percentual de Dropout
GRU	Relu	32	
GRU	Relu	32	0.5
GRU	Relu	16	0.5
Dense	Sigmoid	1	

Tabela 5 - Configuração hiperparâmetros do modelo GRU. Fonte: Elaborado pelo autor.

6.2.2 Long Short Term Memory

Assim como em GRU, o primeiro passo foi definir o estilo arquitetural de LSTM, que pode ser visto na tabela 5. Feito isso, iniciamos os treinos variando a janela de tempo entre 1 e 6 segundos, o número de neurônios e as camadas com o objetivo de encontrar um melhor tempo de treino e aprendizagem do modelo. Os treinos foram executados com 500 épocas e o tamanho do *batch* em 300. Durante os treinos, percebemos que, quando adicionamos um maior número de camadas e neurônios, o modelo aprende mais rápido; no entanto, não percebemos uma melhora nos resultados. Isso pode ser explicado pela quantidade de dados do conjunto de dados UniverCemig.

A Figura 22 apresenta o resultado de treino com a janela de tempo configurada em 6 segundos, o modelo de treino e a validação convergiram para o mesmo valor próximo da época 90. Quando comparado com o resultado do treino de GRU, percebemos que este modelo aprendeu mais rápido; no entanto, utilizou mais poder computacional, aumentou o tempo de treino e não apresentou diferença nos resultados para o conjunto de dados.

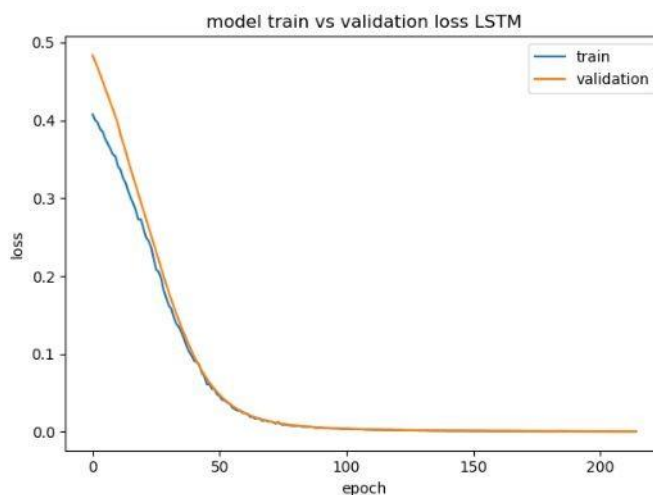
Figura 22 – Modelo de Treino Long Short Term Memory

Figura 22 - Treinamento do modelo LSTM apresentando a variação da validação e treino ao longo das épocas. Fonte: Elaborado pelo autor.

Tabela 6 – Configuração Modelo LSTM e Hiperparâmetros

Hiperparâmetros LSTM			
Camada	Função de Ativação	Número de Neurónios	Percentual de Dropout
LSTM	Relu	16	
LSTM	Relu	16	0.2
LSTM	Relu	8	0.5
Dense	Relu	100	
Dense	Sigmoid	1	

Tabela 6 - Configuração hiperparâmetros do modelo GRU. Fonte: Elaborado pelo autor.

6.2.3 Análise dos Resultados Aprendizado de Máquina Profundo

Nesta seção, apresentaremos os resultados da aplicação dos modelos de AMP, LSTM e GRU. Conforme pode ser observado na Tabela 6, alcançamos precisões de aproximadamente 96% com LSTM e GRU quando tivemos um valor de 6 segundos de janela de tempo.

Quando comparamos os resultados apresentado pelos modelos, percebemos que não existe uma diferença expressiva; no entanto, LSTM utilizou mais recursos computacionais em termos de camada e neurônios, implicando em um aprendizado mais rápido, mas não refletido em melhores resultados.

Tabela 7 – Resultados das Técnicas de SVM e KNN

Janela em Segundos(S)	Algoritmo	Acurácia	Revocação	Precisão
6(S)	LSTM	0.9681	1.0	1.0
	GRU	0.9681	1.0	1.0
5(S)	LSTM	0.8051	1.0	1.0
	GRU	0.8053	1.0	1.0
4(S)	LSTM	0.6971	1.0	1.0
	GRU	0.6972	1.0	1.0
3(S)	LSTM	0.7725	1.0	1.0
	GRU	0.7725	1.0	1.0
2(S)	LSTM	0.5493	0.50	0.4140
	GRU	0.5493	0.3532	0.3532
1(S)	LSTM	0.3514	0.3334	0.1171
	GRU	0.3513	0.3332	0.1171

Tabela 7 - Resultados das técnicas LSTM e GRU de acordo com cada janela de tempo definida. Fonte: Elaborado pelo autor.

A arquitetura de GRU utilizou menos recursos computacionais, menor número de camadas e neurônios; porém, não teve seu resultado alterado. Os dois modelos aplicados suportaram uma maior quantidade de dados, o que era esperado, conforme Aguilera et al. (2019). No entanto, a janela de dados para se alcançasse bons resultados foi de 6 segundos, o que também pode ser explicado devido à quantidade de dados no conjunto de dados UniverCemig. Para ambos os modelos, variamos as funções de ativação nas camadas de entrada e saída, a função de ativação Relu apresentou uma melhor acurácia nas camadas de entrada e, na camada de saída, a função Sigmoid foi a que apresentou a melhor performance.

6.2.3.1 Discussão dos Resultados Aprendizado de Máquina Profundo GRU

O presente trabalho ao aplicar GRU teve como referência na implementação, o trabalho proposto por Gumaei et al. (2019). Em seu projeto, foi desenvolvido um *framework* para realizar o reconhecimento de atividades humanas baseada em multi-sensores usando um modelo de Aprendizado de Máquina Profundo Híbrido, combinando Simple Recurrent Unit e Gated Recurrent Unit. Seus resultados foram comparados com modelos tradicionais de AM utilizando o conjunto de dados público MHEALTH e superaram os principais modelos de Aprendizado de Máquina em uma média de 9%. O modelo do *framework* proposto pode ser visto na tabela 8.

Tabela 8 – Modelo de *framework* proposto por Gumaei et al. (2019)

Hiperparâmetros GRU			
Camada	Função de Ativação	Número de Neurônios	Percentual de Dropout
SRU	Relu	64	0.1
SRU	Relu	64	0.1
GRU	Relu	32	0.1
GRU	Sigmoid	32	0.1

Tabela 8 - Modelo de *framework* proposto por Gumaei et al. (2019). Fonte: Elaborado pelo autor.

Os resultados foram obtidos a partir de dois experimentos: o primeiro utilizou as técnicas de validação Hold-Out, já o segundo utilizou-se técnica de validação cruzada. Na primeira técnica, os dados foram divididos em treino e teste, sendo 70% e 30% respectivamente. Já no segundo experimento, os dados foram divididos em 10 subconjuntos de dados, cada um contendo aproximadamente 336 instâncias. Na primeira abordagem, foi atingida uma acurácia de aproximadamente 94.14% em uma janela de tempo de 1.7 segundo de dados, enquanto a segunda atingiu uma acurácia de 99.80 em uma janela de tempo 2.4 segundos.

Quando comparados os resultados do trabalho proposto por Gumaei et al. (2019) com o presente trabalho, percebemos que os resultados apresentados no presente trabalho estão dentro do esperado, sendo superado em médias em 3%. No entanto, a partir das tabelas de 5 e 8 é possível perceber que o nosso modelo utilizou menos

recursos computacionais e variou também a forma de treinamento, como pode ser visto no item 6.2.1.

O modelo proposto no por este trabalho pode ser destacado por sua simplicidade, embora a janela de tempo para o alcance de um melhor resultado tenha sido de 6 segundos contra 2.4 segundos de dados. Esse fato pode ser explicado pela menor quantidade de instâncias do nosso conjunto de dados, pois segundo Aguilera et al. (2019), modelos de redes neurais tendem a apresentar melhores resultados a partir de grande conjunto de dados.

Analisando os resultados é possível perceber que nosso modelo somente com GRU aproveitou o aprendizado da dinâmica de ativação de recurso temporal, podendo ser indicado para aplicação sobre o conjunto de dados UniverCemig para o reconhecimento de atividades humanas, contribuindo com a aplicação da técnica de AMP em um conjunto de dados criado com atividades de um electricista executando o treinamento de suas atividades cotidianas.

6.2.3.2 Discussão dos Resultados Aprendizado de Máquina Profundo LSTM

A implementação da técnicas LSTM teve como princípio norteador o trabalho proposto por Ordonez and Roggen (2016). O trabalho propôs a criação de um *framework* combinando a arquitetura de LSTM e CNN para o reconhecimento de atividades humanas utilizando dois conjuntos de dados públicos: OPPORTUNITY e SKODA. Os modelos propostos foram treinados de forma supervisionada, tendo seus hiperparâmetros alterados até que se obtivesse o melhor resultado. A arquitetura proposta pode ser visto na Figura 23.

Figura 23 – Modelo de Arquitetura Híbrida

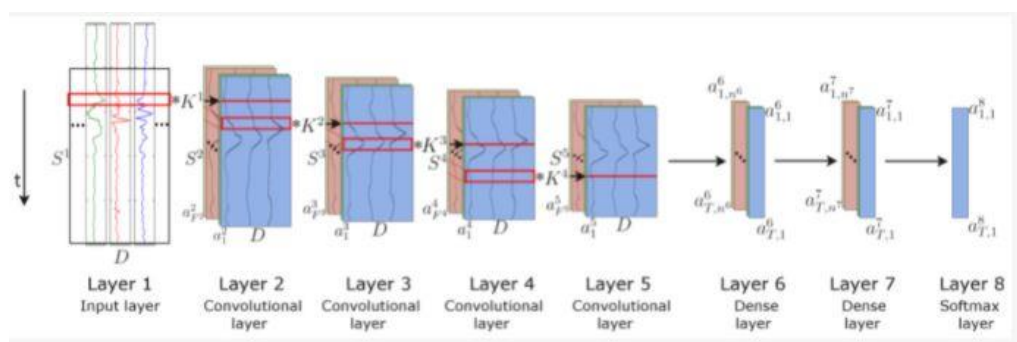


Figura 23 – Modelo de arquitetura Híbrida CNN e LSTM Fonte: Ordonez and Roggen (2016).

Durante a fase de treinamento, foram utilizados subconjuntos de dados com um

tamanho de 100 segmentações. O resultado desse trabalho foi comparado a diversas pesquisas para o reconhecimento de atividades humanas que utilizaram os conjuntos de dados OPPORTUNITY e SKODA. O modelo proposto por Ordonez and Roggen (2016) teve, em média, um aumento de 9% de acurácia em relação as implementações sobre o conjunto de dados OPPORTUNITY, com resultado de 93% de acurácia. Já quando comparado com resultados sobre os sobre trabalhos que utilizaram o conjunto de dados SKODA, essa média cai para 6% com um resultado de 95% de acurácia. De acordo com Ordonez and Roggen (2016), esses resultados só puderam ser alcançados com uso de LSTM, o que traz uma vantagem significativa em cenários diversos.

O modelo proposto neste trabalho pode ser destacado por alcançar resultados de 96% de acurácia em uma janela de tempo de 6 segundos, já que simplifica o modelo e é capaz de compreender as dinâmicas do movimento humano, podendo assim ser indicado para aplicação sobre o conjunto de dados UniverCemig para o reconhecimento de atividades humanas.

7 CONCLUSÃO

O presente teve como objetivos realizar a coleta de dados de alunos em treinamento oferecido pela UniverCemig, utilizando-os para implementar técnicas de aprendizado de máquina e aprendizado de máquina profundo para reconhecer as atividades desempenhadas por alunos durante o processo de capacitação e realizar a análise sobre os resultados. O primeiro objetivo foi atingido com a coleta de dados por meio da filmagem do treinamento de 4 alunos, gerando um vídeo de 1:03:55 horas que, após ser processado, resultou em aproximadamente 98.588 mil instâncias. Os dados extraídos nos permitiram atingir o segundo objetivo, aplicando-os em técnicas de aprendizado de máquina como KNN e SVM e de aprendizado de máquina profundo, como LSTM e GRU, para classificar as atividades executadas.

O terceiro objetivo foi atingido com a análise dos resultados que apontaram que a aplicação de técnicas de aprendizagem profunda apresentaram boa performance quanto aos resultados quando o tamanho da janela supera 6 segundos. Para a execução de trabalhos futuros, faz-se necessário coletar uma maior quantidade de dados e aplicá-los às técnicas com o objetivo de atingir uma boa performance e diminuir a janela de tempo.

Constatamos que as técnicas SVM e KNN demonstraram excelentes resultados mesmo quando a janela de tempo é maior, o que pode ser indicado para a aplicação com o conjunto de dados UniverCemig. A técnica SVM apresentou excelentes resultados em todas as variações de janelas.

Acreditamos que a pesquisa atual pode trazer uma oportunidade de implantação no reconhecimento de atividades humanas ao treinamento dos alunos na UniverCemig, auxiliando os instrutores durante todo o processo e, conseqüentemente, levando agilidade à certificação de aprendizado do aluno. Os resultados experimentais apresentados mostram a eficácia dos modelos propostos e a possibilidade de utilizá-los como *benchmark* para trabalhos futuros utilizando o conjunto de dados proposto.

BIBLIOGRAFIA

Aguilera, A., Brena, R., Mayora, O., Molino Minero Re, E., and Trejo, L. (2019). *Multi-sensor fusion for activity recognition—a survey*. *Sensors*, 19.

Baptista, S. and Cunha, M. (2007). *Estudo de usuários: visão global dos métodos de coleta de dados*. *Perspectivas em Ciência da Informação*, 12.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. *Proceedings*.

Córdova, F. P. and Silveira, D. T. (2008). *Métodos e técnicas de pesquisa social*. <http://197.249.65.74:8080/biblioteca/handle/123456789/707>.

Danaei-mehr, H. and Polat, H. (2019). *Human activity recognition in smart home with deep learning approach*. In *Human Activity Recognition in Smart Home With Deep Learning Approach*, pages 149–153.

Davila, J., Cretu, A.-M., and Zaremba, M. (2017). *Wearable sensor data classification for human activity recognition based on an iterative learning framework*. *Sensors*, 17(6):1287.

Dehghani, A., Sarbishei, O., Glatard, T., and Shihab, E. (2019). *A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors*. *Sensors*, 19:5026.

Filintisis, P., Efthymiou, N., Koutras, P., Potamianos, G., and Maragos, P. (2019). *Fusing body posture with facial expressions for joint recognition of affect in child–robot interaction*. *IEEE Robotics and Automation Letters*, PP:1–1.

Gil (2009). *A pesquisa científica*. in: *Métodos de pesquisa*. Porto Alegre: Editora da UFRGS.

Gil-Martín, M., Sánchez-Hernández, M., and San-Segundo, R. (2019). *Human activity recognition based on deep learning techniques*. *Proceedings*, 42(1):15.

Gumaei, A., Hassan, M. M., Alelaiwi, A., and Alsalman, H. (2019). *A hybrid deep learning model for human activity recognition using multimodal body sensing data*. *IEEE Access*, 7:99152–99160.

Haykin, S. (2001). *REDES NEURAIIS Principios e prática*. 2007, 2 edition.

Hochreiter, S. and Schmidhuber, J. (1997). *Long short-term memory*. *Neural computation*, 9:1735–80.

Hur, T., Bang, J., Huynh-The, T., Lee, J., Kim, J.-I., and Lee, S. (2018). *Iss2image: A novel signal-encoding technique for cnn-based human activity recognition*. *Sensors*, 18:3910.

Huynh, T., Fritz, M., and Schiele, B. (2008). *Discovery of activity patterns using topic models*. In *UbiComp*, pages 10–19.

James L. Crowley, H. I. C. (1995). *Vision as process: basic research on computer vision systems* / J. L. Crowley, H. I. Christensen (eds.); with editorial assistance by: Alain Chehikian e. a., volume 1, pages 341–371. Berlin: Springer, 1995.

Janidarmian, M., Roshan Fekr, A., Radecka, K., and Zilic, Z. (1995). *Support-vector networks*. *machine learning*. page 273–297.

Janidarmian, M., Roshan Fekr, A., Radecka, K., and Zilic, Z. (2017). *A comprehensive analysis on wearable acceleration sensors in human activity recognition*. *Sensors*, 17:529.

Lara, O. D. and Labrador, M. A. (2013). *A survey on human activity recognition using wearable sensors*. *IEEE Communications Surveys Tutorials*, 15(3):1192–1209.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). *Deep learning*. *Nature*, 521:436–44.

Lei, T., Zhang, Y., Wang, S., Dai, H., and Artzi, Y. (2018). *Simple recurrent units for highly parallelizable recurrence*. pages 4470–4481.

Long, J., Sun, W., Yang, Z., and Raymond, O. I. (2019). *Asymmetric residual neural network for accurate human activity recognition*. *Information*, 10(6):203.

Manzi, A., Dario, P., and Cavallo, F. (2017). *A human activity recognition system based on dynamic clustering of skeleton data*. *Sensors*, 17(5):1100.

McCulloch, W. S. and Pitts, W. (1943). *A logical calculus of the ideas immanent in nervous activity*. *The bulletin of mathematical biophysics*. Cambridge, Massachusetts, USA, 1 edition.

Mitchell, R., Michalski, J., and Carbonell, T. (2013). *An artificial intelligence approach*. Cambridge, Massachusetts, USA, 1 edition.

Nguyen, P., Akiyama, T., Ohashi, H., Nakahara, G., Yamasaki, K., and Hikaru, S. (2015). *User-friendly activity recognition using svm classifier and informative features*. pages 1–8.

Ordonez, F. and Roggen, D. (2016). *Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition*. *Sensors*, 16(1):115.

Pardos, A., Menychtas, A., and Maglogiannis, I. (2020). *Introducing an Edge-Native Deep Learning Platform for Exergames*, pages 88–98.

Poppe, R. (2010). *A survey on vision-based human action recognition*. *Image and Vision Computing*, 28(6):976 – 990.

Prodanov, C. C. and C.Freitas, E. (2013). *Metodologia do trabalho científico: Métodos e técnicas da pesquisa e do trabalho acadêmico*. Rio Grande do Sul: Universidade

FEEVALE.

Roggen, D., Calatroni, A., Rossi, M., Holleczeck, T., Forster, K., Troster, G., Lukowicz, P., Bannach, D., Pirkl, G., Ferscha, A., Doppler, J., Holzmann, C., Kurz, M., Holl, G., Chavarriaga, R., Sagha, H., Bayati, H., Creatura, M., and Millan, J. d. R. (2010). *Collecting complex activity datasets in highly rich networked sensor environments*. pages 233 – 240.

Rosati, S., Balestra, G., and Knaflitz, M. (2018). *Comparison of different sets of features for human activity recognition by wearable sensors*. *Sensors*, 18(12):4189.

Saccol, A. (2010). *Um retorno ao básico: compreendendo os paradigmas de pesquisa e sua aplicação na pesquisa em administração*. *Revista de Administração da UFSM*, 2(2):250–269.

Sáez, Y., Gómez, A. B., and Viñuela, P. I. (2016). *A comparison study of classifier algorithms for cross-person physical activity recognition*. In *Sensors*.

Sapiński, T., Kamińska, D., Pelikant, A., and Anbarjafari, G. (2019). *Emotion recognition from skeletal movements*. *Entropy*, 21:646.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). *Real-time human pose recognition in parts from single depth images*. In *CVPR 2011*, pages 1297–1304.

Straka, M., Hauswiesner, S., Rüdter, M., and Bischof, H. (2011). *Skeletal graph based human pose estimation in real-time*.

Su, X., Tong, H., and Ji, P. (2014). *Activity recognition with smartphone sensors*.

Tsinghua Science and Technology, 19:235–249.

Sugiyama, M. and Kawanabe, M. (2012). *Machine Learning in Non-Stationary*

Environments: Introduction to Covariate Shift Adaptation, volume 4 of 10. The name of the publisher, Cambridge, MA, USA, 1 edition.

Tasnim, N., Islam, M. M., and Baek, J.-H. (2020). *Deep learning-based action recognition using 3d skeleton joints information*. *Inventions*, 5(3):49.

Tuceryan, M. and Jain, A. (2015). *Handbook of Pattern Recognition and Computer Vision: 5th Edition*, volume 5, page 582.

Vahora, S. and Chauhan, N. (2019). *Deep neural network model for group activity recognition using contextual relationship*. *Engineering Science and Technology, an International Journal*, 22(1):47–54.

Viswambaran, R. A., Chen, G., Xue, B., and Nekooei, M. (2019). *Evolutionary design of recurrent neural network architecture for human activity recognition*. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 554–561.

Wang, Y., Velswamy, K., and Huang, B. (2017). *A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems*. *Processes*, 5(3).

Yan, S., Xiong, Y., and Lin, D. (2018). *Spatial temporal graph convolutional networks for skeleton-based action recognition*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Yu, T., Chen, J., Yan, N., and Liu, X. (2018). *A multi-layer parallel lstm network for human activity recognition with smartphone sensors*. pages 1–6.

Zebin, T., Peek, N., Casson, A., and Sperrin, M. (2018). *Human activity recognition from inertial sensor time-series using batch normalized deep lstm recurrent networks*. volume 2018.

Kwapisz, J.R.; Weiss, G.M.; Moore, S.A. *Activity recognition using cell phone*

accelerometers. *ACM SigKDD Explor. Newsl.* 2011, 12, 74–82.

Bachlin, M.; Plotnik, M.; Roggen, D.; Maidan, I.; Hausdorff, J.M.; Giladi, N.; Troster, G. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Trans. Inf. Technol. Biomed.* 2010, 14, 436–446