



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE SOBRAL
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

MARIA ELINE SILVA DE FARIAS

**EXPLORANDO OS MOVIMENTOS DAS MÃOS COMO INTERFACE PARA
CONTROLAR COMPUTADORES: UM PROTÓTIPO UTILIZANDO O
MICROCONTROLADOR ESP32**

SOBRAL

2024

MARIA ELINE SILVA DE FARIAS

EXPLORANDO OS MOVIMENTOS DAS MÃOS COMO INTERFACE PARA CONTROLAR
COMPUTADORES: UM PROTÓTIPO UTILIZANDO O MICROCONTROLADOR ESP32

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do *Campus* Sobral da Universidade Federal do Ceará, como requisito parcial para obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Carlos Alexandre Rolim Fernandes.

SOBRAL

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

F238e Farias, Maria Eline Silva de.
EXPLORANDO OS MOVIMENTOS DAS MÃOS COMO INTERFACE PARA CONTROLAR
COMPUTADORES: UM PROTÓTIPO UTILIZANDO O MICROCONTROLADOR ESP32 / Maria Eline
Silva de Farias. – 2024.
52 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral,
Curso de Engenharia da Computação, Sobral, 2024.
Orientação: Prof. Dr. Carlos Alexandre Rolim Fernandes..

1. Reconhecimento de gestos das mãos. 2. Aprendizado de máquina. 3. ESP32. 4. Acelerômetro. 5.
Classificação. I. Título.

CDD 621.39

MARIA ELINE SILVA DE FARIAS

EXPLORANDO OS MOVIMENTOS DAS MÃOS COMO INTERFACE PARA CONTROLAR
COMPUTADORES: UM PROTÓTIPO UTILIZANDO O MICROCONTROLADOR ESP32

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do *Campus* Sobral da Universidade Federal do Ceará, como requisito parcial para obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: 18 de Setembro de 2024

BANCA EXAMINADORA

Prof. Dr. Carlos Alexandre Rolim
Fernandes (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Iális Cavalcante de Paula Júnior
Universidade Federal do Ceará (UFC)

Prof. Me. David Nascimento Coelho
Universidade Federal do Ceará (UFC)

Para as mulheres que batalham diariamente pelo direito de serem vistas como pessoas em um mundo que insiste em ignorar sua existência.

AGRADECIMENTOS

Este TCC reflete o apoio fundamental de todas as pessoas que fizeram parte da minha experiência universitária.

Em primeiro lugar, gostaria de expressar minha sincera gratidão à minha família, que sempre me incentivou na busca pela educação. Acreditando sempre que somente através dela poderíamos transformar nossa realidade.

Agradeço a cada amizade que formei e pelos momentos vividos ao lado de vocês, que se transformou em uma lembrança preciosa. Nossas conversas foram fundamentais, tornando cada desafio mais fácil de enfrentar. Principalmente, minhas amigas Lais Gomes e Maxela Martins, que me acompanham desde o início nessa universidade e trouxeram tanta alegria e apoio nessa caminhada.

Agradeço especialmente ao meu orientador, Prof. Dr. Carlos Alexandre Rolim Fernandes, cuja orientação paciente e dedicada foi essencial ao longo deste trabalho. Seus ensinamentos representaram uma fonte constante de aprendizado para mim.

Ao Me. Francisco José Prado Júnior e ao Me. Flávio Vasconcelos dos Santos pelo valioso apoio técnico fornecido no início do projeto e à aluna de Engenharia de Computação Lislândia Bastos de Lima, pela sua dedicação no desenvolvimento do designer do projeto.

Aos professores participantes da banca examinadora, Prof. Me. David Nascimento Coelho e Prof. Dr. Iális Cavalcante de Paula Júnior pelo tempo e pelas valiosas sugestões.

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro através da minha bolsa de Iniciação Científica, possibilitando a elaboração deste trabalho.

"The best way to learn about something is doing it. "

(Maria Carolina Monard)

RESUMO

Este trabalho apresenta o desenvolvimento do protótipo de um sistema para o reconhecimento de gestos das mãos como interface para controlar computadores. O dispositivo utiliza sensores MPU-6050 para a captura dos dados de movimento, sendo então processados por um microcontrolador ESP32. Para o acoplamento dos componentes, foram confeccionadas peças personalizadas em uma impressora 3D, abordagem escolhida para tentar garantir o máximo de conforto e flexibilidade para os usuários. O processo de coleta de dados envolveu quatro usuários realizando seis diferentes movimentos das mãos, resultando em um conjunto de dados composto por 1.440 gestos. Diversos classificadores foram avaliados para determinar a eficácia do sistema, incluindo *Random Forest (RF)*, *Multilayer Perceptron (MLP)*, *Extreme Gradient Boosting (XGBOOST)* e *Support Vector Machine (SVM)*. O classificador SVM, configurado com *kernel Radial Basis Function (RBF)* e com a aplicação da *Principal Component Analysis (PCA)*, alcançou uma acurácia média de 95,14%, indicando uma boa capacidade de generalização do modelo e a eficácia da redução de dimensionalidade em melhorar o desempenho. Os resultados obtidos demonstram o potencial do sistema desenvolvido para o controle de computadores baseados em gestos, oferecendo uma alternativa promissora para interações humano-computador mais intuitivas e acessíveis.

Palavras-chave: reconhecimento de gestos das mãos; ESP32; acelerômetro; giroscópio; aprendizado de máquina; classificação.

ABSTRACT

This work presents the development of a prototype system for gesture recognition as an interface to control computers. The device utilizes MPU-6050 sensors to capture motion data, which is then processed by an ESP32 microcontroller. Customized components were crafted using a 3D printer to assemble the device, a method chosen to ensure maximum comfort and flexibility for users. The data collection process involved four users performing six different hand movements, resulting in a dataset composed of 1,440 gestures. Various classifiers were evaluated to determine the system's effectiveness, including Random Forest (RF), Multilayer Perceptron (MLP), Extreme Gradient Boosting (XGBOOST), and Support Vector Machine (SVM). The SVM classifier, configured with a Radial Basis Function (RBF) kernel and with the application of Principal Component Analysis (PCA), achieved an average accuracy of 95.14%, indicating a good generalization capability of the model and the effectiveness of dimensionality reduction in improving performance. The results obtained demonstrate the potential of the developed system for gesture-based computer control, offering a promising alternative for more intuitive and accessible human-computer interactions.

Keywords: hand gesture recognition; ESP32; accelerometer; gyroscope; machine learning; classification.

LISTA DE FIGURAS

Figura 1 – MPU-6050	17
Figura 2 – Diagrama de blocos	18
Figura 3 – Kernel aplicado em duas classes	21
Figura 4 – Representação gráfica de uma RNA-MLP com duas camadas intermediárias	22
Figura 5 – PCA aplicado em um conjunto de dados	24
Figura 6 – Matriz de confusão	25
Figura 7 – Esquemático do circuito do sistema de captação de gestos.	32
Figura 8 – Sistema a ser modelado	33
Figura 9 – Sistema completo construído.	33
Figura 10 – Sistema fixado na mão do usuário	34
Figura 11 – Posição inicial da mão para os movimentos.	36
Figura 12 – Movimentos de mão escolhidos.	37
Figura 13 – Resultado dos testes utilizando 30 componentes de PCA.	42
Figura 14 – Comparação dos tempos de execução dos classificadores com e sem PCA. .	44
Figura 15 – Matriz de confusão - SVM com PCA (%).	45
Figura 16 – Matriz de confusão com 4 classes - SVM com PCA (%).	48

LISTA DE TABELAS

Tabela 1 – Dados dos participantes	34
Tabela 2 – Resumo das características da base de dados	35
Tabela 3 – Melhores hiperparâmetros para os classificadores	41
Tabela 4 – Acurácias e desvios padrão dos classificadores em 20 iterações	46
Tabela 5 – Métricas de desempenho para cada classe	46

LISTA DE ABREVIATURAS E SIGLAS

I ² C	<i>Inter-Integrated Circuit</i>
ADC	<i>Analog-to-Digital Converter</i>
AM	Aprendizado de Máquina
CNN	<i>Convolutional Neural Network</i>
DAC	<i>Digital-to-Analog Converter</i>
FFT	<i>Fast Fourier Transform</i>
GPIO	<i>General Purpose Input/Output</i>
IA	Inteligência Artificial
IDE	Ambiente de Desenvolvimento Integrado
IoT	<i>Internet of Things</i>
MEMS	<i>Micro Electro Mechanical System</i>
MLP	<i>Multilayer Perceptron</i>
PCA	<i>Principal Component Analysis</i>
RBF	<i>Radial Basis Function</i>
RF	<i>Random Forest</i>
RP	Reconhecimento de Padrões
SNR	<i>Signal-to-Noise</i>
SVM	<i>Support Vector Machine</i>
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
USB	<i>Universal Serial Bus</i>
XGBOOST	<i>Extreme Gradient Boosting</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Contexto e Motivação	12
1.2	Contribuições	14
1.3	Estrutura do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Sensores aplicados ao reconhecimento de gestos	16
2.2	ESP32: um microcontrolador aplicado em sistemas de detecção de gestos	17
2.3	Aprendizado de Máquina	19
2.3.1	<i>Support Vector Machine</i>	20
2.3.2	<i>Multilayer Perceptron</i>	21
2.3.3	<i>Floresta Aleatória</i>	22
2.3.4	<i>Extreme Gradient Boosting</i>	23
2.3.5	<i>Principal Component Analysis</i>	23
2.3.6	<i>Avaliação de modelos de aprendizado de máquina</i>	24
3	REVISÃO BIBLIOGRÁFICA	27
4	SISTEMA DE AQUISIÇÃO PROPOSTO	31
4.1	Sistema para captação dos dados	31
4.2	Construção da base de dados	34
4.2.1	<i>Definição dos Tipos de Movimentos</i>	36
4.3	Extração de Atributos	37
4.4	Classificação	40
5	RESULTADOS	41
5.1	Seleção dos hiperparâmetros	41
5.2	Análise Geral dos Resultados da Classificação	42
5.3	Comparação dos Tempos de Execução	43
5.4	Análise do melhor caso	44
5.4.1	<i>Análise do melhor caso com redução para 4 classes</i>	47
6	CONCLUSÕES E TRABALHOS FUTUROS	49
	REFERÊNCIAS	50

1 INTRODUÇÃO

1.1 Contexto e Motivação

A partir da década de 1950, o campo da Inteligência Artificial (IA) tem passado por períodos de estagnações e avanços, conhecidos como “invernos e verões” da IA (RUSSELL; NORVIG, 2002). Nos últimos anos, a IA está vivenciando um período de verão intenso, atribuído principalmente ao aumento substancial do poder de processamento computacional e ao grande volume de dados. Esse momento foi gerado especialmente pela disseminação das redes sociais e pela integração de dispositivos de *Internet of Things* (IoT) no cotidiano de grande parte da sociedade. Essa abundância de dados fornece a matéria-prima necessária para a evolução dos algoritmos de Aprendizado de Máquina (AM), um dos subcampos da IA (AGGARWAL; YU, 2021), que mais tem se desenvolvido.

Allende-Cid (2019) define o AM como uma classe de ferramentas composta por algoritmos matemáticos, estatísticos e computacionais que possibilitam a realização de processos de inferência por meio do aprendizado baseado em exemplos. Essa característica do AM, faz dele uma ferramenta poderosa para diversas aplicações, como, por exemplo, em plataformas de *streaming* que possuem serviço de recomendação (WANG; ZHANG, 2018), nas filtragens de conteúdo em redes sociais, em sistemas que transcrevem áudio em texto e no reconhecimento e agrupamento de fotografias semelhantes, em *softwares* de gerenciamento de imagens (TAPPEN *et al.*, 2017).

Além das aplicações citadas acima, uma das áreas mais relevantes e amplamente aplicáveis do AM é o Reconhecimento de Padrões (RP). Este consiste na aplicação de algoritmos e técnicas de AM com o intuito de reconhecer padrões em conjuntos de dados. Os sistemas de RP seguem três etapas principais: na primeira etapa ocorre a aquisição das informações de entrada. Na segunda etapa, ocorre a extração e pré-processamento dos atributos, como filtros e, caso necessário, a redução de dimensionalidade. Na terceira etapa, o classificador ou descritor é projetado (MARQUES, 1999; THEODORIDIS; KOUTROUBAS, 2009; TOU, 2018).

A evolução da miniaturização dos componentes eletrônicos fez surgir conceitos como o de “Computação vestível” e “tecnologia vestível” que se referem a uma nova abordagem da computação, onde o objetivo é tornar a comunicação entre homem e máquina mais simples e intuitiva (KUMAR *et al.*, 2012). É nesse contexto que o reconhecimento de gestos, uma das subáreas do campo do RP, aparece como uma solução inovadora, uma vez que constitui

uma maneira natural, intuitiva e conveniente de interação entre diferentes meios (XIE; CAO, 2016). O desenvolvimento da computação vestível e o reconhecimento de gestos oferecem uma flexibilidade maior, permitindo que os usuários executem várias tarefas sem estarem limitados a esses dispositivos tradicionais, onde as interfaces de interação com dispositivos eletrônicos são específicas e estáticas, projetadas para serem controladas por teclados, mouses e controles remotos.

Em Mitra e Acharya (2007) é proposta a divisão de gestos em três categorias: gestos corporais, expressões faciais e gestos dos membros superiores. Em relação à última categoria proposta, o reconhecimento de gestos das mãos é a etapa inicial para um computador entender a linguagem corporal humana, desempenhando um papel-base em uma ampla gama de aplicações de interação humano-computador. Duas aplicações principais para tecnologias de reconhecimento de gestos com as mãos são o reconhecimento de linguagem de sinais e o controle baseado em gestos (MITRA; ACHARYA, 2007).

O reconhecimento das linguagens de sinais visa interpretar automaticamente as línguas de sinais por meio de um computador, com o objetivo de facilitar a comunicação das pessoas surdas com a sociedade ouvinte (ZHANG *et al.*, 2011). Além disso, o *mouse* tem sido amplamente utilizado para o controle de máquinas, entretanto, para pessoas com deficiências físicas ou para aquelas que não são habituadas a seu uso, vão enfrentar limitações (AL-SAEDI; AL-ASADI, 2019).

Com base no que foi exposto acima, este trabalho apresenta um protótipo de um dispositivo de reconhecimento de gestos de movimentos de mão. Toda a estrutura do sistema foi cuidadosamente planejada com o intuito de garantir que os custos fossem mantidos no mínimo possível. O sistema funciona com um microcontrolador ESP32, responsável pelo controle e processamento dos dados, capturados por três sensores MPU-6050.

O desenvolvimento deste dispositivo incluiu várias etapas, começando com a seleção dos componentes eletrônicos apropriados. A escolha dos sensores MPU-6050 se deve à sua capacidade de medir tanto aceleração quanto orientação, fornecendo dados detalhados e precisos sobre os movimentos da mão. O microcontrolador ESP32 foi selecionado por sua eficiência e capacidades de processamento adequadas para o projeto.

A estrutura mecânica do dispositivo foi projetada para maximizar o conforto e a funcionalidade. Utilizando a tecnologia de impressão 3D, foram criadas peças personalizadas para a fixação dos sensores e do microcontrolador na mão do participante. Essa abordagem

permitiu ajustes precisos que consideram a ergonomia e a necessidade de manter os sensores firmemente fixados durante a utilização, sem comprometer a liberdade de movimento dos usuários. Além disso, a escolha por peças impressas em 3D também traz vantagens significativas em termos de durabilidade.

Uma base de dados foi desenvolvida utilizando esse sistema, contendo informações de quatro usuários, com um total de 1440 movimentos distribuídos em seis classes distintas. Para garantir a eficácia do dispositivo, foram realizadas várias simulações e testes com diferentes tipos de classificadores e atributos. O objetivo foi otimizar a precisão do sistema e identificar a melhor taxa de acerto possível.

1.2 Contribuições

As principais contribuições deste trabalho são listadas a seguir:

- Projeto do Sistema de Captação de Dados: Elaboração de um projeto detalhado para o desenvolvimento do sistema de captação de gestos, incluindo a definição dos componentes necessários, escolha dos sensores e microcontrolador, e planejamento do *design* estrutural das peças de fixação dos componentes do sistema, levando em consideração aspectos como estabilidade dos sensores e principalmente conforto do usuário.
- Desenvolvimento de um sistema de reconhecimento de gestos, que realiza a captação de gestos da mão.
- Criação de um banco de dados contendo 1440 gestos de mãos realizados por quatro participantes.
- Realização de várias simulações, variando os diversos tipos de classificadores, técnicas de classificação e tipos de atributos para melhorar a precisão do sistema e determinar a melhor taxa de acerto.

1.3 Estrutura do Trabalho

O restante do trabalho está organizado como se segue. No Capítulo 2 é apresentada a fundamentação teórica, essencial para a compreensão das tecnologias utilizadas neste trabalho. No Capítulo 3, é realizada uma revisão bibliográfica, visando explorar as contribuições existentes na literatura sobre o assunto abordado. Essa revisão é essencial para a contextualização do trabalho e para o embasamento teórico utilizado durante as análises e discussões dos resultados

obtidos. No Capítulo 4 é explicado como foi construído e funciona o sistema de captação. No Capítulo 5, são apresentados os resultados obtidos na classificação. Por fim, no Capítulo 6, são apresentadas as conclusões deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são discutidos os fundamentos teóricos necessários para compreender o Trabalho de Conclusão de Curso. A Seção 2.1 aborda os principais sensores empregados no processo de reconhecimento de gestos. A Seção 2.2 discorre sobre as especificações relevantes relacionadas ao ESP32. Por fim, a Seção 2.3 trata das principais técnicas adotadas na área de AM.

2.1 Sensores aplicados ao reconhecimento de gestos

Para o reconhecimento de gestos, é fundamental a utilização de sensores apropriados. Dentre os sensores empregados nesse contexto, destacam-se câmeras, sensores de profundidade, sensores de movimento, acelerômetros e giroscópios.

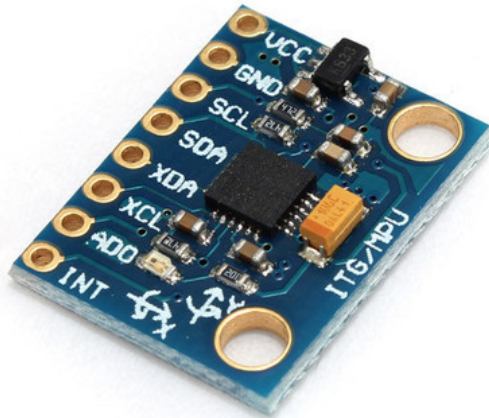
É esperado que os dispositivos utilizados para a captura de movimentos sejam não invasivos e apresentem uma relação *Signal-to-Noise* (SNR) elevada, a fim de possibilitar uma extração adequada de atributos. Alguns estudos de reconhecimento gestual, por exemplo, empregam sensores de eletromiografia (EMG) para a captura de sinais (NAVARRO; VARGAS, 2010). Esses sensores são particularmente relevantes, dado que possibilitam que indivíduos que tenham sofrido perda parcial da mão os utilizem com facilidade, além de não causarem desconforto ou interferência ao usuário. No entanto, é importante destacar que esses sensores possuem uma SNR relativamente baixa em comparação a outras categorias de sensores (FRANCISCO *et al.*, 2020).

Outra abordagem utilizada atualmente no reconhecimento de gestos recorre a múltiplos sistemas de câmeras, os quais acarretam custos expressivos e demonstram uma notável sensibilidade às flutuações de iluminação, posicionamento da mão e alterações ambientais. Esses fatores representam desafios significativos para a precisão e robustez desses sistemas, limitando a sua eficácia em cenários práticos e variados.

Um sensor versátil, economicamente viável e aplicável no contexto de reconhecimento de gestos é o MPU-6050, conforme ilustrado na Figura 1. Esse dispositivo é composto por um giroscópio de três eixos (x, y, z) e um acelerômetro de três eixos (x, y, z), totalizando um conjunto de 6 graus de liberdade. Utilizando a tecnologia *Micro Electro Mechanical System* (MEMS), o giroscópio do MPU6050 detecta a velocidade de rotação ao longo dos eixos X, Y e Z. Já o acelerômetro, também com tecnologia MEMS, mede o ângulo de inclinação ou a inclinação ao longo desses mesmos eixos (INVENSENSE, 2013). Esses seis canais de comunicação

fornecem informações distintas, as quais podem ser processadas e empregadas em algoritmos de classificação.

Figura 1 – MPU-6050



Fonte: INVENSENSE (2013).

A comunicação com o MPU-6050 se estabelece por meio do padrão *Inter-Integrated Circuit* (I²C), por meio dos pinos de comunicação *serial clock* (SCL) e *serial data* (SDA). Uma vantagem adicional desse sensor reside na sua capacidade de permitir o uso de vários dispositivos em um único microcontrolador, graças ao controle de endereço disponível na porta de entrada AD0. Essa funcionalidade proporciona a flexibilização da disposição dos sensores em diferentes configurações, resultando na otimização da obtenção de informações.

O MPU-6050 digitaliza os valores dos eixos x, y e z coletados dos giroscópios e acelerômetros por meio de conversores analógico/digital (A/D). Além disso, os valores lidos são armazenados em registradores internos, onde o tamanho de cada um desses registradores internos é de 8 bits, e a precisão dos giroscópios e acelerômetros é de 16 bits. Portanto, cada valor coletado é armazenado em dois registradores internos, um para os 8 bits mais significativos e o outro para os 8 bits menos significativos (QUIRINO *et al.*, 2018).

2.2 ESP32: um microcontrolador aplicado em sistemas de detecção de gestos

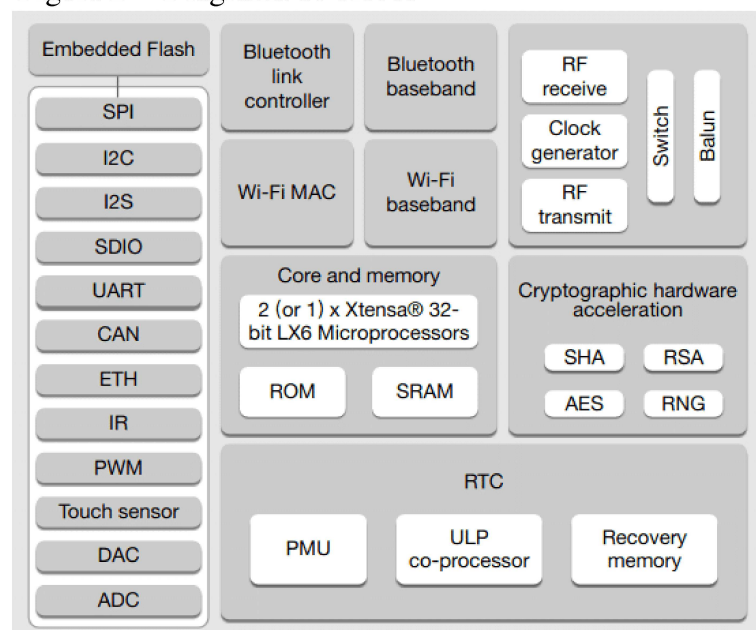
Uma das formas de realizar a comunicação de interface entre os dispositivos e o computador consiste no emprego de microcontroladores. Esses dispositivos, compostos por circuitos integrados, são dotados de uma unidade lógica e aritmética, unidade de controle,

memórias, periféricos de entrada e saída, temporizadores, além de portas de comunicação serial e outros elementos (DESHMUKH, 2005).

A seleção do microcontrolador desempenha um papel crucial na implementação do sistema desenvolvido, e a seleção adequada é fundamental para garantir o desempenho, funcionalidade e eficiência do dispositivo proposto. Existem vários microcontroladores utilizados na indústria e em projetos de engenharia, como o ESP32, uma série de microcontroladores de sistema em chip de baixo custo e baixo consumo de energia, desenvolvido pela empresa chinesa Espressif Systems. O ESP32 é um sucessor avançado do chip 8266, principalmente devido à implementação de dois núcleos com *clock* em diferentes versões de até 240 MHz. O chip ESP32 é disponibilizado no mercado em várias versões, como o ESP32 *Developer Kit*, o ESP32 *Wrover Kit* e o ESP32 *Azure IoT kit* (BABIUCH *et al.*, 2019).

Na Figura 2 é mostrado o diagrama de bloco e funções do ESP32, com sua estrutura projetada para operar sob os protocolos *TCP/IP*, *WLAN* e *MAC WLAN*. Esse microcontrolador possui capacidades de *Wi-Fi* e *Bluetooth* e pode operar no modo de energia ativo e no modo de hibernação do modem. Os *General Purpose Input/Output (GPIO)* incluem dois *Analog-to-Digital Converter (ADC)* de 12 bits com um total de 18 canais e dois *Digital-to-Analog Converter (DAC)* de 8 bits. Além disso, o ESP32 possui várias interfaces: uma Interface *MAC Ethernet*, três interfaces *Universal Asynchronous Receiver-Transmitter (UART)* de até 5 Mbps e duas interfaces de barramento *I²C* (MAIER *et al.*, 2017).

Figura 2 – Diagrama de blocos



Fonte: Andrade (2022).

O ESP32 pode ser programado em nos sistemas operacionais *Windows, Linux ou MacOS*. Além disso, temos a possibilidade de trabalhar com o ambiente adicional ESP32 na IDE do Arduino, o que foi a opção neste trabalho devido à sua simplicidade de configuração, permitindo uma prototipagem rápida. Também é possível utilizar a plataforma nativa *Espressif IoT Development Framework* e o ambiente *Python* usando o *Micropython*. Esse microcontrolador suporta várias linguagens de programação como *C, C++, Python, Lua, MicroPython e JavaScript*. Outra característica é seu baixo custo e alto desempenho que garantem captura e processamento eficiente de dados, exemplificado pela comunicação com sensores MPU-6050 para reconhecimento de gestos em tempo real.

Devido a todas essas características, o ESP32 torna-se uma opção versátil e ideal para uma ampla gama de aplicações, incluindo sistemas embarcados, utilizados na construção de casas inteligentes, automação, *wearables*, aplicações de áudio e aplicações de IoT baseadas em nuvem.

2.3 Aprendizado de Máquina

O AM é reconhecido como uma das tecnologias mais transformadoras da atualidade. Esse campo abrange a detecção automática de padrões e a análise de dados (THANTILAGE *et al.*, 2023). Segundo Mahadevkar *et al.* (2022), os principais tipos de AM são o aprendizado supervisionado, não supervisionado e por reforço.

No aprendizado supervisionado, cada amostra (dado) fornecido ao algoritmo inclui a resposta desejada, ou seja, um rótulo que indica a classe a que o exemplo pertence. Esse método é utilizado em tarefas de classificação, onde cada amostra é representada por um vetor de valores (atributos) e pelo rótulo da classe associada. No aprendizado não supervisionado, as amostras são fornecidas ao algoritmo sem rótulos. O algoritmo analisa as amostras fornecidas e tenta determinar se algumas delas podem ser agrupadas de alguma maneira, formando os *clusters* (LUDERMIR, 2021).

Por fim, no aprendizado por reforço, os agentes (algoritmos de AM ou sistemas autônomos) são treinados de forma a ajustarem seu processo de tomada de decisão em função de reforços recebidos (recompensa ou punição) pelas ações que escolheram ao longo de sua atuação (RUSSELL; NORVIG, 2002).

Classificação, regressão e agrupamento são tarefas comuns realizadas com algoritmos de AM. Para garantir uma boa execução dessas tarefas, é fundamental atender a certos pré-

requisitos. É necessário dispor de um conjunto de dados adequados; devido à complexidade desse requisito, frequentemente é necessário empregar técnicas para aprimorar a qualidade dos dados. A seleção do algoritmo é outro pré-requisito fundamental, e, após essa escolha, é crucial definir seus parâmetros e avaliar, após a fase de treinamento, se o algoritmo está resolvendo o problema e com que precisão (LUDERMIR, 2021).

Nos próximos tópicos, será apresentada uma fundamentação teórica concisa para cada técnica que será utilizada neste trabalho de conclusão de curso (TCC), abordando seus princípios e formas de avaliação.

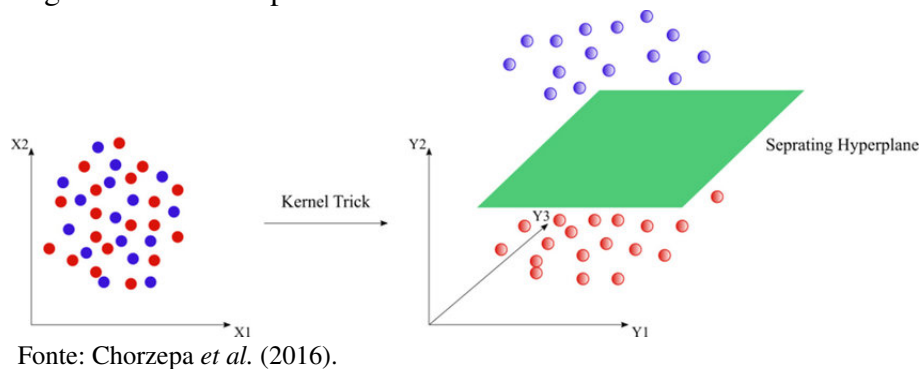
2.3.1 *Support Vector Machine*

O SVM é um algoritmo de aprendizado supervisionado, desenvolvido por Cortes e Vapnik (1995), amplamente utilizado na resolução de problemas de classificação e regressão. É aplicado em vários campos, incluindo análise de imagens e processamento de linguagem natural. Analisando a sua aplicabilidade em problemas de classificação, o SVM é baseado em um conceito fundamental chamado de hiperplano de separação. Um hiperplano é uma superfície que divide um espaço de dados em duas partes, de forma que as amostras de uma classe fiquem de um lado do hiperplano e as amostras da outra classe fiquem do outro lado. O objetivo do SVM é encontrar o hiperplano de separação ótimo, que maximize a margem entre as duas classes. A margem é a distância entre o hiperplano e as amostras de treinamento mais próximas.

Uma das principais vantagens do SVM é a capacidade de lidar com dados não linearmente separáveis. Para isso, o SVM utiliza uma técnica chamada de “*kernel trick*”. Essa técnica mapeia os dados de entrada para um espaço de maior dimensionalidade, onde eles podem ser linearmente separáveis. Isso é feito por uma função de *kernel*, que calcula o produto interno entre dois vetores no espaço de maior dimensionalidade, sem a necessidade de calcular explicitamente as coordenadas desse espaço. Esta técnica permite ao SVM identificar um hiperplano ótimo, mesmo quando os dados não são linearmente separáveis no seu espaço original. A Figura 3 ilustra a transformação de um domínio não linearmente separável, em um problema linearmente separável através do aumento da dimensão, onde é feito um mapeamento por uma função de *kernel*.

Apesar do SVM apresentar um bom poder de generalização (ABE, 2005), seu desempenho depende da seleção dos parâmetros na função de *Kernel* do classificador. A escolha de parâmetros inadequados pode interferir na acurácia dos resultados. Algumas das funções

Figura 3 – Kernel aplicado em duas classes



de *kernel* comumente utilizados no SVM são o *Kernel Linear*, usado no SVM linear, onde as classes são linearmente separáveis, além dos *Kernels*, Polinomial e *Radial Basis Function (RBF)*, empregados quando não se consegue uma separação linear dos dados.

2.3.2 *Multilayer Perceptron*

A rede neural MLP é uma das técnicas mais empregadas no campo do AM. Ela é composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada camada contém um conjunto de neurônios interconectados. A camada de entrada é responsável por receber os dados iniciais fornecidos à rede neural, propagados através das camadas ocultas até a camada de saída.

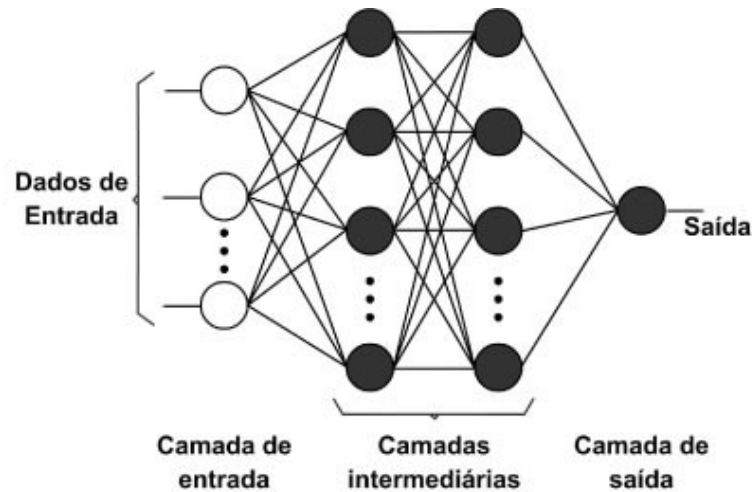
Determinar o número de camadas e neurônios em uma MLP é um processo empírico que depende da complexidade do problema, quantidade de dados de treinamento, presença de ruído nos dados e distribuição estatística dos mesmos. A escolha da arquitetura adequada envolve experimentação e ajuste para encontrar a configuração que melhor se adapte às características específicas do problema.

Cada neurônio em um MLP realiza uma combinação linear das entradas, aplicando uma função de ativação para gerar uma saída. A função de ativação é geralmente não-linear, como a função sigmoideal ou a função tangente hiperbólica, permitindo ao MLP modelar relações não-lineares complexas entre os dados de entrada e a saída desejada.

Na Figura 4 é mostrado um exemplo de MLP com três entradas, duas camadas intermediárias com quatro neurônios e uma camada de saída com um neurônio, produzindo uma única informação final.

A MLP é amplamente utilizada em diversas aplicações, como reconhecimento de padrões, processamento de linguagem natural, visão computacional e previsão de séries temporais.

Figura 4 – Representação gráfica de uma RNA-MLP com duas camadas intermediárias



Fonte: Fiorin *et al.* (2011).

Sua propriedade de modelar relações complexas entre os dados torna-o uma ferramenta poderosa no AM.

2.3.3 Floresta Aleatória

O aprendizado por agrupamento ou por comitê (*Ensemble Learning*) é uma classe de métodos de AM que treina múltiplos modelos mais simples (*weak learners*) para resolver o mesmo problema e os combina para obter um modelo agrupado melhor (*strong learner*). Na aplicação desses métodos, os *weak learners* podem ser, por exemplo, árvores de decisão, redes neurais, SVM, ou outros algoritmos de aprendizado (WANG *et al.*, 2019).

A floresta aleatória (RF) é um algoritmo de AM que utiliza *Ensemble Learning*, criando um conjunto de várias árvores de decisão de modo aleatório, para obter uma predição estável e com maior acurácia (BREIMAN, 2001). É também um dos algoritmos mais utilizados, devido à sua simplicidade, flexibilidade e ser menos propenso a *overfitting*, podendo ser aplicado tanto em tarefas de classificação quanto de regressão.

As árvores de decisão, as componentes-base do RF, são modelos hierárquicos de aprendizado supervisionado, formado por nós de decisão e folhas terminais. Cada nó de decisão aplica um teste aos dados de entrada, de modo a seguir por um dos ramos que levará a um novo nó de decisão ou a uma folha terminal, onde é determinado o valor da variável de interesse (ALPAYDIN, 2010).

As árvores no RF são criadas extraindo um subconjunto de amostras de treinamento

com reposição, conhecida como abordagem de *bagging*. Isso significa que a mesma amostra pode ser selecionada várias vezes, enquanto outras podem não ser escolhidas. Essa estratégia ajuda a reduzir o *overfitting*, um problema comum em modelos de AM.

Os principais hiperparâmetros do classificador RF são o número de árvores, o critério de divisão, que define as métricas de pureza do nó, permitindo que o algoritmo realize divisões que proporcionem uma separação mais eficiente dos dados, e a profundidade máxima da árvore (SILVA, 2021).

Em problemas de classificação, RF usa um mecanismo de votação para determinar a previsão final. Cada árvore de decisão no conjunto fornece sua própria previsão de classe, e a classe que recebe a maioria dos votos entre as árvores é selecionada como a classe prevista final.

2.3.4 *Extreme Gradient Boosting*

O XGBOOST, proposto por Chen e Guestrin (2016), é um algoritmo de AM que também implementa o método *Ensemble Learning* com um mecanismo de aumento de gradiente chamado *Gradient Boosting*. O *Boosting* consiste em construir modelos sequenciais onde o resultado do primeiro modelo passa para o próximo com o intuito de reduzir erros e favorecer modelos de alto desempenho até que uma condição de parada seja alcançada. O aumento de gradiente consiste em minimizar erros em modelos sequenciais (XGBOOST, 2022).

O XGBOOST usa como *weak learner* árvores de decisão e vem apresentando um forte impacto na comunidade de AM devido às melhorias em relação ao RF (PEREIRA, 2021). Essas melhorias incluem árvores mais eficientes, minimizando erros e, por isso, mais resistente ao ruído e desequilíbrio de classe. Como resultado, o XGBOOST vem se destacando em relação ao seu desempenho em tarefas de classificação (SAMAT *et al.*, 2020).

2.3.5 *Principal Component Analysis*

PCA teve sua primeira descrição realizada por Pearson em 1901. No entanto, somente com as contribuições de Hotelling em 1933 e 1936, que uma descrição mais detalhada dos métodos computacionais práticos foi apresentada.

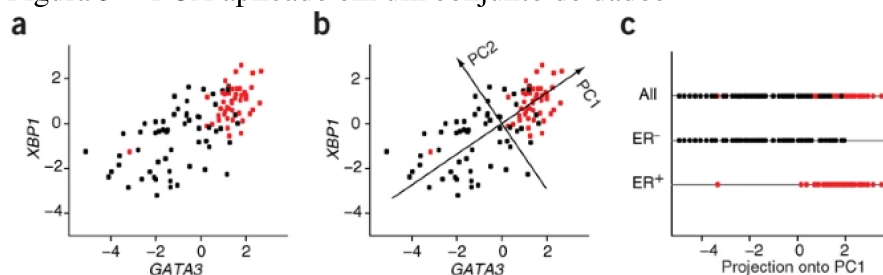
PCA é um método de redução de dimensionalidade frequentemente utilizado para transformar grandes conjuntos de dados complexos em conjuntos menores, preservando as informações mais importantes. Embora a redução do número de dados resulte em alguma perda de precisão, o objetivo é equilibrar essa perda com a simplicidade (JAADI, 2021). Essa redução

de dimensionalidade ocorre identificando novos eixos, conhecidos como componentes principais, que são variáveis construídas como combinações lineares das variáveis iniciais (RINGNER, 2008). Além disso, esses componentes são ortogonais entre si, ou seja, não são correlacionados.

Antes de aplicar a técnica PCA, primeiro é essencial padronizar os dados, isso garante que todas as variáveis tenham igual importância na análise. Em seguida, calcula-se a matriz de covariância. A covariância entre duas variáveis mostra o quanto elas mudam juntas, dessa forma, o objetivo desta etapa é entender como as variáveis do conjunto de dados de entrada estão variando da média em relação umas às outras.

Após, são calculados os autovalores e autovetores da matriz de covariância, onde os autovalores indicam a variância contida em cada autovetor (componente principal). Ao ordenar os autovetores conforme a relevância de seus autovalores, são identificados os componentes principais. Esses autovetores ortogonais representam a maior variação nos dados, formando uma matriz de transformação para projetar os dados originais em um espaço de menor dimensão, reduzindo assim o conjunto de dados.

Figura 5 – PCA aplicado em um conjunto de dados



Fonte: adaptado de Ringner (2008).

Na Figura 5a cada ponto representa uma amostra. Já na Figura 5b o PCA identifica as duas direções (PC1 e PC2) ao longo das quais os dados apresentam a maior dispersão e, por fim, na Figura 5c as amostras são plotadas em uma dimensão utilizando suas projeções no primeiro componente principal (PC1).

2.3.6 Avaliação de modelos de aprendizado de máquina

A etapa de avaliação dos modelos de AM permite determinar o quão bem o modelo está realizando a tarefa para a qual foi projetado. Essa etapa fornece métricas e medidas objetivas para comparar diferentes modelos e abordagens, auxiliando na identificação daquelas que são mais eficazes. Com base nos resultados da avaliação, é possível realizar ajustes e aprimoramentos no modelo, como modificar sua arquitetura, ajustar hiperparâmetros ou adotar técnicas de pré-

processamento dos dados.

É importante que um modelo não apenas se ajuste aos dados de treinamento, mas também possa lidar com dados nunca vistos. Nesse sentido, a avaliação pode fornecer *insights* sobre a capacidade do modelo de generalizar e evitar o *overfitting*. Existem diversas abordagens e métricas que podem ser utilizadas para avaliar um modelo AM, e sua escolha depende de fatores como a natureza do problema, os dados disponíveis e os objetivos específicos do projeto.

Para avaliar os modelos, primeiramente deve-se particionar os dados em conjuntos disjuntos de treinamento e validação. Dentre as abordagens utilizadas, estão a *hold-out*, onde os dados são divididos em conjuntos de treinamento e teste. Outra abordagem é a *k-fold (cross-validation)*, que divide os dados em *k* partes (*folds*). O processo envolve realizar *k* iterações, onde em cada iteração um dos *folds* é usado como conjunto de teste e os *k-1 folds* restantes são usados para treinamento. A média das métricas de desempenho das *k* iterações é então calculada, oferecendo uma avaliação mais precisa do modelo ao utilizar diferentes divisões dos dados. Já o *leave-one-out* é uma forma especial da *k-fold*, onde cada amostra é usada uma vez como teste, proporcionando uma avaliação detalhada do modelo.

Cada uma dessas abordagens tem suas vantagens e desvantagens. E a escolha mais adequada depende do tamanho do conjunto de dados, da disponibilidade de recursos computacionais e do grau de relevância, da precisão da avaliação do modelo.

A matriz de confusão é uma ferramenta visual que detalha o desempenho do modelo. Ela consiste em quatro características básicas usadas para definir as métricas de medição do classificador.

Figura 6 – Matriz de confusão

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Rodrigues (2019)

Essas características são: Verdadeiros Positivos, que representam a classificação correta da classe Positivo; Falsos Negativos, que indicam onde o modelo previu a classe Negativo quando o valor real era classe Positivo; Falsos Positivos, que mostram o erro em que o modelo

previu a classe Positivo quando o valor real era classe Negativo; e Verdadeiros Negativos, que refletem a classificação correta da classe Negativo, conforme ilustrado na Figura 6.

No contexto de problemas de classificação, as métricas mais utilizadas na literatura são: a acurácia, que fornece uma medida geral do desempenho do classificador, indicando a capacidade de classificar corretamente as amostras. Definida como:

$$Acuracia = \frac{\text{Amostras classificadas corretamente}}{\text{Total de amostras}} . \quad (2.1)$$

A precisão, por sua vez, é uma métrica que avalia a proporção de amostras classificadas como positivas que são realmente positivas. É útil quando o objetivo está em minimizar os falsos positivos, ou seja, reduzir as classificações incorretas de exemplos negativos como positivos. Seu cálculo é da seguinte forma:

$$Precisao = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos positivos}} . \quad (2.2)$$

Sensibilidade (*Recall*) representa a medida que, dentre todas as previsões positivas feitas, quantas de fato estão corretas. É importante em situações onde os Falsos Negativos são considerados mais prejudiciais que os Falsos Positivos. Definida como:

$$Sensibilidade = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}} . \quad (2.3)$$

F1 Score combina precisão e sensibilidade para obter um único valor que indique a qualidade geral do modelo. É bastante utilizado em conjunto de dados que possuem classes desproporcionais. A fórmula que define o *F1 Score* é a seguinte:

$$F1Score = \frac{2 \cdot \text{Precisao} \cdot \text{Recall}}{\text{Precisao} + \text{Recall}} . \quad (2.4)$$

Especificidade avalia a capacidade do método de detectar resultados negativos. Dada por:

$$Especificidade = \frac{\text{Verdadeiros negativos}}{\text{Verdadeiros negativos} + \text{Falsos positivos}} . \quad (2.5)$$

3 REVISÃO BIBLIOGRÁFICA

O reconhecimento de gestos refere-se à capacidade de extrair informações relevantes dos movimentos realizados pelo usuário, para interação com outros dispositivos (RYLO, 2023). A literatura mostra diversas técnicas utilizadas para realizar o reconhecimento e classificação de gestos, que depende dos recursos disponíveis e da viabilidade da aplicação.

Nos trabalhos que aplicam visão computacional, são utilizadas técnicas de processamento de imagens para identificação e classificação de gestos. Essa abordagem é usada em Alon *et al.* (2009) por meio de técnicas de segmentação espacial e temporal que se baseiam em estimativas da posição e movimento da mão. Nesse estudo, a precisão foi de até 85% e a aplicação foi testada no reconhecimento contínuo de dígitos em tempo real e na identificação de sinais da língua de sinais americana em um banco de dados de vídeos.

Em Patil *et al.* (2022), uma *webcam* capta a imagem dos gestos detectados com métodos como segmentação da área da pele, usada para separar a imagem da mão do fundo e a técnica de AM cascata de *haar* para detecção de objetos. Já Reddy *et al.* (2020) empregam uma abordagem semelhante com tampas coloridas fixadas nas pontas dos dedos durante a captura da imagem pela *webcam*. Para o processamento das imagens nesse caso, são utilizados algoritmos de subtração de fundo e identificação de cores, além de técnicas de contorno e cascas convexas. Ambos os estudos possuem como aplicação o controle do mouse com diferentes gestos da mão.

Misra e mmwaeli (2010) propõem um sistema de reconhecimento de gestos baseado na análise do brilho nos blocos da imagem segmentada, usando o modelo de cor que representa cores por três componentes: Matiz (Hue), Saturação (Saturation) e Valor (Value). A imagem é dividida em blocos e o brilho de cada bloco é calculado, resultando em um vetor de características. O sistema alcançou uma taxa de reconhecimento de 91%. Já em Sharma *et al.* (2019), imagens de gestos são capturadas e salvas em um vídeo. O reconhecimento dos gestos é realizado considerando as características como pontas dos dedos e ângulos entre os dedos. O sistema alcançou uma precisão de 95,2% no reconhecimento de gestos de mão.

Hussain *et al.* (2017) sugerem, para o reconhecimento das formas da mão, um classificador baseado em *Convolutional Neural Network* (CNN), o qual é treinado usando transferência de aprendizado. Gestos são rastreados para detectar movimento, utilizando algoritmos de extração de centroides. Experimentos mostraram uma precisão de 93,09%.

Lee-Ferng *et al.* (2009) descrevem um sistema de reconhecimento de gestos de mão para interação em tempo real com robôs de serviço em ambientes dinâmicos. O sistema

reconhece gestos por estatísticas temporais das posições e velocidades das mãos e utiliza um classificador de Bayes. O sistema teve uma taxa de reconhecimento de gestos de 75%.

As abordagens que se utilizam de uma ou mais câmeras como forma de captura de gesto estão sujeitas a certos obstáculos, como fundos complexos, presença de objetos com cor parecida com a cor da pele da mão, desfoque de movimentos, baixa resolução no vídeo ou imagem. Além disso, possuem limitações na redução do tamanho dos sistemas criados e são sensíveis a ambientes de iluminação externa (KIM *et al.*, 2017).

Outra abordagem que está sendo bastante explorada é a utilização de sensores para mapeamento dos gestos. Prado (2020), Francisco *et al.* (2020) e Navarro e Vargas (2010) usam sinais eletromiográficos (EMG), com os eletrodos posicionados no antebraço. A classificação é com SVM, alcançando uma taxa de identificação de até 97% em Prado (2020), 91.34% em Francisco *et al.* (2020) e 99,25% em Navarro e Vargas (2010). Além disso, Prado (2020) e Francisco *et al.* (2020) aplicam o sistema de reconhecimento de gestos da mão na criação de um editor de texto adaptado.

Kim *et al.* (2008) também usam sensores EMG posicionados na parte interna do antebraço. Para realizar a classificação, são combinados dois classificadores, k-NN e Bayes, tendo taxa de precisão de 94%. O sistema foi aplicado no controle com gestos de mãos de um carro de controle remoto.

Outros sensores bastante utilizados nos estudos atuais são os sensores que possuem acelerômetros e giroscópios, uma vez que fornecem informações necessárias para a decomposição do movimento (BONATTO, 2017). Essa abordagem é aplicada em Xie e Cao (2016), que apresentam um dispositivo de reconhecimento de gestos, onde os gestos são realizados por um sistema em forma de caneta e que possui um acelerômetro. As acelerações geradas pelos movimentos das mãos são transmitidas para um computador via cabo USB. Ainda nesse estudo, é desenvolvido um algoritmo de reconhecimento de gestos manuais baseado em Rede Neural *feedforward* e correspondência de similaridade e atinge-se uma taxa de precisão de 98,88% para os gestos complexos.

Outro trabalho onde o sistema é em formato de caneta está presente em Hsu *et al.* (2015), sendo diferenciado do anterior pela forma de transmissão que é sem fio e adição do valor da leitura do magnetômetro. Além disso, usa-se um algoritmo de reconhecimento baseado em *dynamic time warping (DTW)* para reconhecimentos de dígitos e letras em inglês manuscritas 2D com taxas de reconhecimento dependentes de 99,8%.

Gupta *et al.* (2016) usam os giroscópios e acelerômetros presentes no *smartphone*, com um algoritmo de alinhamento temporal dinâmico, para reconhecer os gestos realizados com o *smartphone*, comparando o código do gesto com um banco de dados de gestos e a precisão média deste experimento é de 94%. Já em (CORREIA *et al.*, 2021) foram utilizados sensores contendo acelerômetros e giroscópios acoplados à mão direita do indivíduo, depois é aplicado o algoritmo KMeans para segmentação dos gestos e para a classificação é utilizada a técnica de RF, atingindo uma acurácia de 83%.

Kumar *et al.* (2021) propõem um intérprete de linguagem de sinais para surdos-mudos utilizando uma luva equipada com sensores flexíveis e o sensor MPU-6050, conectado a um *Raspberry Pi*. Os sensores flexíveis detectam os ângulos de flexão dos dedos, enquanto o MPU-6050 fornece dados de inclinação e movimento da mão. Os dados são processados pelo *Raspberry Pi*, que, por um programa em *Python*, compara-os com uma base de dados de gestos. O sistema traduz os gestos reconhecidos em texto e áudio, exibindo o resultado em um LCD e emitindo uma mensagem de voz.

Anusha e Devi (2016) apresentam a conversão dos gestos em voz usando um sistema composto com sensor MPU-6050 e *Raspberry Pi*. O MPU-6050 captura os gestos, enquanto o *Raspberry Pi* reconhece os alfabetos por meio de um algoritmo de reconhecimento de trajetória. Nesse trabalho, também é utilizada uma plataforma que permite transformar texto escrito em áudio natural, o *Voice RSS*, que gera a saída de voz do texto reconhecido, já o *Microsoft Translator* traduz a voz para diversos idiomas.

Aponno *et al.* (2022) desenvolvem uma tecnologia assistiva para o Sistema de Interpretação da Língua de Sinais da Indonésia, focada na conversão de sinais do alfabeto em texto. O sistema possui sensores flexíveis, MPU-6050 e de pressão. A classificação utilizou um método de detecção de menor erro, ajustando um dicionário de limites do conversor analógico-digital para cada indivíduo. A precisão desse sistema foi de 94% e, para melhorar o resultado, sugeriu-se a necessidade de personalização das luvas devido às variações nas dimensões das mãos e exploração do uso de métodos de AM.

Santos (2020) e Santos (2023) fixaram dois sensores MPU-6050 em uma luva de algodão. Em Santos (2020), foram testados diversos classificadores, sendo realizada a aplicação de PCA, resultando em uma taxa de acerto de 94,27% ao usar o classificador SVM com seis classes e 10 componentes de PCA. Já em Santos (2023), foram utilizadas funções de kernel tensorial para a etapa de classificação.

Gourob *et al.* (2021) e Carneiro *et al.* (2016) fazem uso tanto de sensores de profundidade Kinect, quanto de técnicas de processamento de imagens. No primeiro, gestos são capturados em tempo real por uma câmera e suas imagens são processados com a biblioteca OpenCV para identificar pontos de convexidade que determinam os gestos, os quais geram comandos binários para um microcontrolador, controlando um braço robótico. No segundo, sinais da Língua Brasileira de Sinais são reconhecidos usando a técnica *eigenhands*, a qual é adaptada da PCA, que representa imagens de mãos como vetores de alta dimensionalidade e classifica-as pela distância euclidiana em relação aos *eigenvectors* treinados.

Embora exista uma vasta literatura na área de reconhecimento de gestos, essas abordagens oferecem uma variedade de perspectivas e soluções inovadoras para os desafios enfrentados na interação humano-computador. À medida que a tecnologia continua a evoluir, é fundamental explorar essas diferentes abordagens para criar sistemas de reconhecimento de gestos mais precisos, acessíveis e adaptáveis.

Nesse contexto de constante inovação, o trabalho desenvolvido se diferencia dos estudos revisados pela utilização da impressão 3D para a criação de peças personalizadas. A combinação do microcontrolador ESP32 com os sensores MPU-6050, aliada à flexibilidade da impressão 3D, permitiu a construção de um sistema mais acessível. Além disso, é utilizada a extração de atributos nos domínios do tempo, frequência e *Wavelet*, o que possibilitou uma caracterização mais completa dos gestos.

4 SISTEMA DE AQUISIÇÃO PROPOSTO

Neste capítulo, é apresentado o sistema de captação e classificação de movimentos de mão desenvolvido. O objetivo foi criar um sistema de baixo custo de produção, que forneça uma boa mobilidade ao usuário e com alta taxa de acerto. Para isso, foi desenvolvido um sistema com sensores de giro e aceleração acoplados em peças feitas em uma impressora 3D, capaz de enviar os dados para o microcontrolador ESP32.

Na Seção 4.1, será detalhado o sistema de captação, incluindo todo o processo de criação e os materiais utilizados. Já na Seção 4.2, serão descritos os processos necessários para a construção do banco de dados, que foi gravado utilizando o sistema de captação desenvolvido. Na Seção 4.3, será feita a descrição dos atributos utilizados na classificação. Por fim, na Seção 4.4, são apresentados os detalhes da etapa de classificação, incluindo os algoritmos e técnicas de AM utilizadas no decorrer deste trabalho.

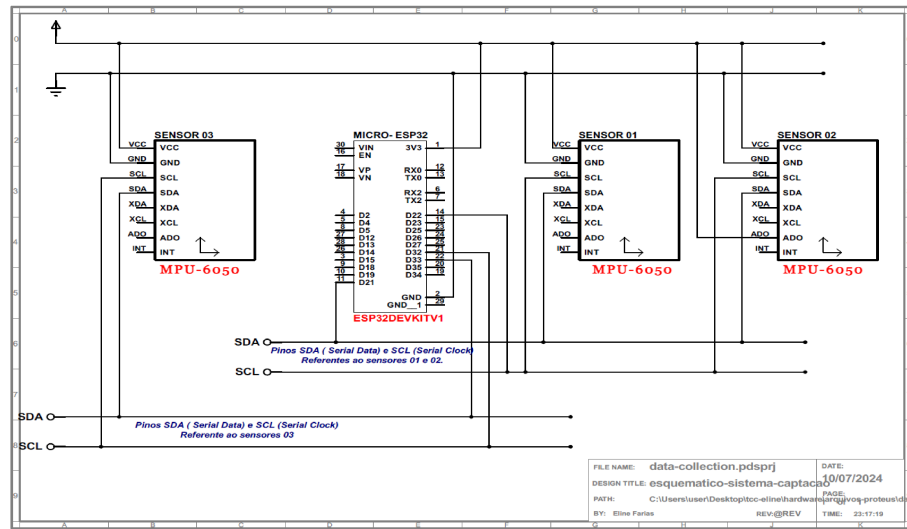
4.1 Sistema para captação dos dados

Visando captar os movimentos das mãos de forma eficiente e desenvolver um sistema de captação de baixo custo que oferecesse ao usuário boa mobilidade durante seu uso, foi feita a escolha por utilizar sensores do tipo MPU-6050 para mapear o movimento das mãos. O MPU-6050 é um sensor versátil e economicamente viável. Além disso, a comunicação com o sensor se estabelece por meio do padrão I²C, permitindo a conexão de vários dispositivos em um único microcontrolador, conforme descrito na Seção 2.1.

Para o processamento e comunicação dos dados coletados pelos sensores, optou-se pelo microcontrolador ESP32. Este microcontrolador é conhecido por seu baixo custo, alta eficiência, capacidade de ser programado em diversas linguagens e oferecer uma excelente relação desempenho/custo, tornando-o ideal para este projeto, conforme fundamentado na Seção 2.2. O esquema elétrico do sistema projetado está ilustrado na Figura 7, tendo sido elaborado utilizando o *software* Proteus.

Após a definição dos componentes, que foram utilizados no sistema, foram testadas duas abordagens para a montagem dos componentes. Inicialmente, considerou-se a opção de utilizar luvas de algodão como base para a fixação dos sensores, tal como em Santos (2023). No entanto, após alguns testes, foi identificado alguns contras em relação ao uso da luva de algodão, entre eles a interferência na precisão dos sensores, amortecendo os movimentos

Figura 7 – Esquemático do circuito do sistema de captação de gestos.



Fonte: elaborada pelo autor (2024).

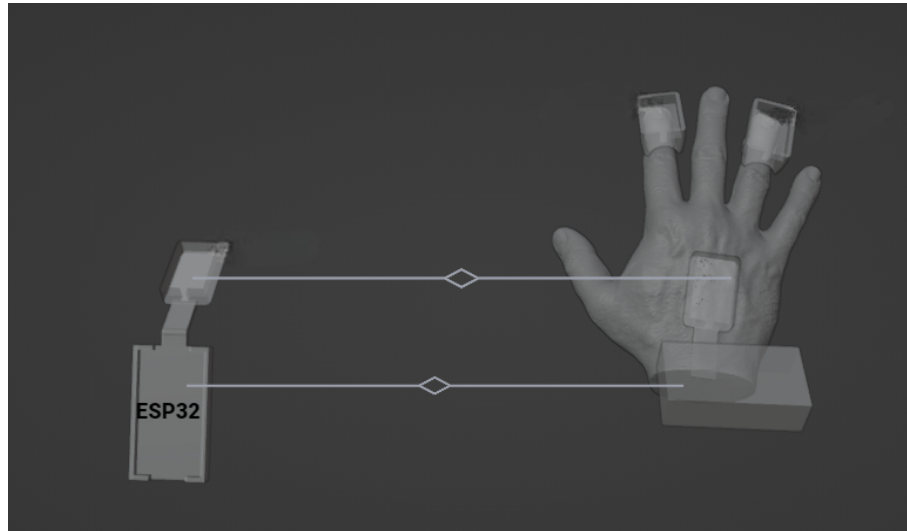
das mãos e dificultando a captura precisa dos gestos. Além disso, a fixação dos sensores e do microcontrolador na luva foi um desafio, por ser necessário garantir que os sensores permanecessem firmes e estáveis durante a interação, sem prejudicar o conforto do usuário.

Diante dessas considerações, optou-se por uma segunda abordagem, através do desenvolvimento de peças personalizadas utilizando uma impressora 3D. As peças foram projetadas em parceria com a *startup* AssistiveTech, que desempenhou um papel fundamental na etapa inicial desse trabalho. Além de fornecer as peças, a AssistiveTech contribuiu significativamente para a concepção da ideia, oferecendo sugestões sobre as melhores formas de projetar as peças para atender às necessidades específicas desse projeto. Essa escolha permitiu a criação de suportes específicos para os sensores e o microcontrolador. Considerando os requisitos de *design*, estabilidade e facilidade de fixação dos sensores e do microcontrolador na mão, foram elaborados protótipos das peças usando o *software* de modelagem 3D *Blender*, ilustrado na Figura 8.

Um sensor está localizado sobre o dorso da mão, por ser uma área central e relativamente estável durante a maioria dos gestos, possibilitando a coleta de dados adicionais sobre a orientação e a aceleração dos movimentos. Este sensor complementa os outros dois, localizados no polegar e no dedo médio. O polegar foi escolhido devido à sua maior flexibilidade, permitindo uma ampla variedade de movimentos. Já o dedo médio foi selecionado por estar localizado no centro da mão, proporcionando uma média representativa dos movimentos dos outros dedos.

Essa distribuição estratégica permite capturar de forma abrangente os movimentos e gestos realizados pela mão durante a interação. Ao combinar as informações dos três sensores, foi possível mapear de forma mais precisa os movimentos da mão no espaço tridimensional,

Figura 8 – Sistema a ser modelado



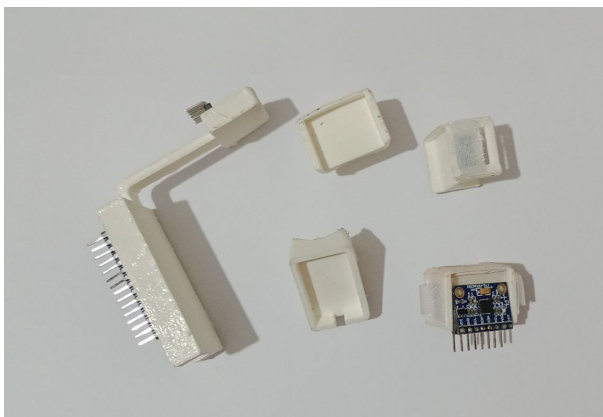
Fonte: adaptada de PRADO, F. J. J, 2023

contribuindo para uma melhor acurácia na interpretação dos gestos e reduzindo a margem de erro, o que aumenta a confiabilidade do dispositivo.

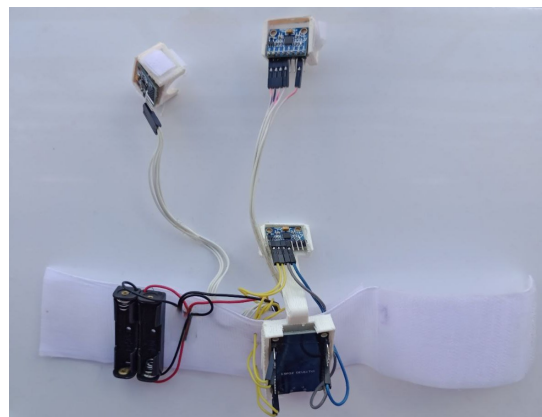
As peças impressas em 3D demonstraram ser mais duráveis e resistentes ao desgaste em comparação com o material de uma luva de algodão. Para sua impressão, foram utilizados o material *PLA Flex*, o que garantiu a resistência e durabilidade necessárias. A Figura 9 ilustra o sistema construído. Na Figura 9a são mostradas todas as peças testadas e na Figura 9b temos o sistema completo com as peças finais escolhidas para compor o sistema.

Figura 9 – Sistema completo construído.

(a) Peças confeccionadas



(b) Sistema completo construído

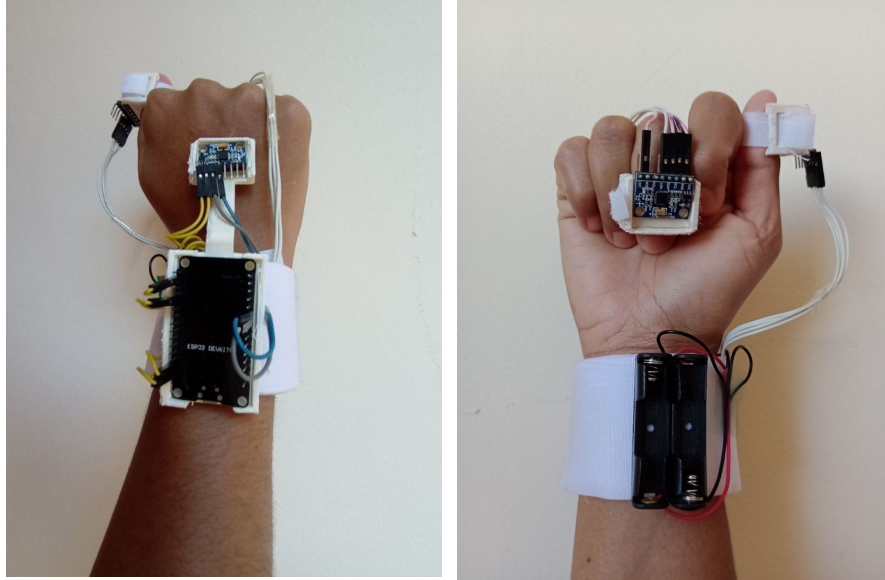


Fonte: elaborada pelo autor (2024).

A Figura 10 ilustra o sistema em uso na mão do usuário. A Figura 10a apresenta a vista do dorso da mão, destacando a localização do sensor 03, enquanto a Figura 10b mostra a vista da palma da mão, evidenciando como o sistema é flexível durante os movimentos realizados

pelo usuário.

Figura 10 – Sistema fixado na mão do usuário
(a) Vista dorso da mão (b) Vista palma da mão



Fonte: elaborada pelo autor (2024).

4.2 Construção da base de dados

Com a conclusão do sistema de captação de gestos, foi possível iniciar a fase de coleta dos dados, cujos dados foram gerados por quatro participantes distintos. Algumas informações sobre os participantes que construíram a base de dados são apresentadas na Tabela 1.

Tabela 1 – Dados dos participantes

Identificação dos participantes	Idade	Sexo
Participante 1	27	M
Participante 2	28	M
Participante 3	34	H
Participante 4	60	M

Fonte: elaborada pelo autor (2023).

A base de dados é composta por 18 colunas, cada uma representando um eixo (canal) de comunicação resultante dos três sensores que compõem o sistema de reconhecimento de gestos. As seis primeiras colunas correspondem ao sensor um, as seis colunas seguintes ao sensor dois, e as últimas seis colunas ao sensor três. Com um total de 388.800 linhas, essa base é estruturada de forma que um bloco de 270 linhas corresponde a um movimento (amostra). A base de dados pode ser acessada a partir do link: [BASE DE DADOS](#).

Para garantir a consistência e a robustez dos dados, cada um dos seis movimentos predefinidos foi rotulado como uma das seis classes estudadas e repetido 60 vezes por cada participante. Assim, para cada classe, foram obtidas 240 amostras. No total, a base de dados contém 1.440 sinais, com uma distribuição uniforme entre as diferentes classes e participantes. A Tabela 2 apresenta um resumo das características da base de dados.

Tabela 2 – Resumo das características da base de dados

características	Quantidade
Número de participantes	4
Total de classes	6
Total de Canais de comunicação	18
Movimentos por classe	60
Pontos coletados por canal	270

Fonte: elaborada pelo autor (2024).

O algoritmo de captação de gestos foi implementado no Ambiente de Desenvolvimento Integrado (IDE) Arduino, portanto o algoritmo é baseado na linguagem de programação C++. Inicialmente, o programa ativa a comunicação serial, possibilitando a comunicação do microcontrolador com o computador utilizando uma conexão *Universal Serial Bus* (USB). Os sensores 1 e 2 compartilham o primeiro barramento I²C, enquanto o sensor 3 utiliza o segundo barramento.

Ao usar o ESP32 com a IDE Arduino, foi permitido usar pinos SDA e SCL diferentes dos padrões e as duas interfaces I²C do ESP32, conforme detalhado na seção 2.2. Com isso, e utilizando a biblioteca *Wire.h*, foi possível distinguir e se comunicar com os três sensores.

Após essas configurações iniciais, o algoritmo realiza uma verificação para assegurar que todos os sensores estejam funcionando corretamente. Com os sensores conectados corretamente, o algoritmo entra em um estado de espera, monitorando o canal $ax[1]$, que corresponde ao valor da aceleração do eixo x do sensor localizado no dedo médio. Quando esse valor sair do intervalo $14000 < ax[1] < 18000$, o algoritmo inicia a coleta de dados.

Para salvar os dados do monitor serial em um arquivo de extensão .txt, um segundo algoritmo foi criado utilizando a linguagem Python 3.12. Esse algoritmo também fica sendo executado durante a captação dos movimentos.

4.2.1 Definição dos Tipos de Movimentos

Os movimentos foram realizados de modo que a força e velocidade de execução fossem variadas entre diferentes níveis, visando permitir que o classificador seja treinado para discernir essas variações na execução dos movimentos. Para garantir uma padronização na captação do movimento, foi estabelecido que a mão do usuário deve estar na posição vertical, conforme ilustrado na Figura 11, e retornar à posição inicial após a execução do movimento.

Figura 11 – Posição inicial da mão para os movimentos.



Fonte: elaborada pelo autor (2024).

A seguir, encontra-se a descrição de cada movimento selecionado:

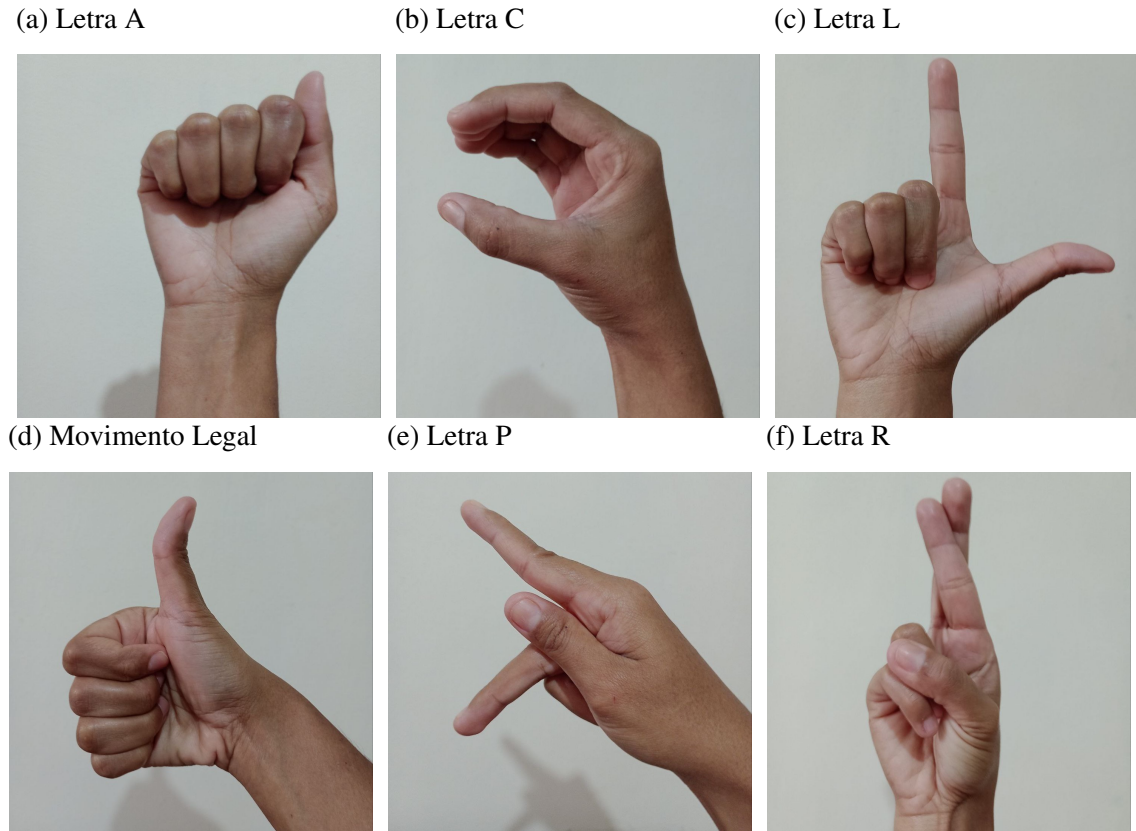
- **Letra A:** Feche a mão e estique o polegar para cima.
- **Letra C:** Forme a letra C com a mão, curvando os dedos e o polegar para criar uma forma semicircular.
- **Letra L:** Estenda o polegar e o indicador formando um ângulo de 90 graus, mantendo os outros dedos fechados.
- **Movimento Legal:** O dedo polegar deve estar apontando para cima, enquanto os outros permanecem fechados.
- **Letra P:** A mão é posicionada com a palma voltada para baixo. O dedo médio é apontado para baixo, enquanto os dedos polegar e indicador formam um ângulo de 90 graus com o

dedo médio. Os outros dois dedos restantes permanecem fechados.

- **Letra R:** Os dedos anelar e mínimo estão fechados, tocando a palma da mão. Já os dedos indicador e médio estão estendidos e cruzados entre si e o dedo polegar fica sobre os dedos fechados.

Na figura 12 são mostrados os movimentos realizados:

Figura 12 – Movimentos de mão escolhidos.



Fonte: elaborada pelo autor (2024).

4.3 Extração de Atributos

A escolha dos atributos em RP tem uma grande influência sobre: a precisão da classificação, o tempo necessário para a classificação, o número de amostras necessárias para o aprendizado e o custo de realizar a classificação (MEYER-BÄSE, 2004). Por isso, para determinar os melhores atributos, foram realizados diversos testes nos domínios do tempo e da frequência e usando componentes de Wavelet. A seguir, é discutido cada atributo utilizado na etapa de classificação:

Atributos no Domínio do Tempo:

- Valor Mínimo: O valor mínimo absoluto do sinal (amostra) é útil para identificar o menor ponto do sinal, ajudando no entendimento da variabilidade do sinal.
- Curtose: Representa o grau de achatamento da distribuição, isto é, o quão a distribuição tem ou não caudas pesadas.
- *Zero Cross Rate*: Determina quantas vezes o sinal cruza o eixo zero. É um recurso importante a ser considerado porque nos dá uma representação da suavidade da onda ao dizer o número de vezes que ela se torna de positiva para negativa e vice-versa (GARCÍA *et al.*, 2022).
- Valor Central: Mediana do sinal. É menos sensível a *outliers* em comparação com a média, proporcionando uma visão mais robusta do centro das amostras.
- Valor Máximo: O valor máximo de um sinal é uma métrica crucial para entender o comportamento do sinal e suas características extremas. Analisando o valor máximo, é possível detectar picos e entender a variabilidade do sinal.
- Assimetria: Trata-se de uma medida amostral que assume o valor zero quando a distribuição é completamente simétrica e assume valores diferentes de zero (positivos ou negativos) quando a distribuição não é simétrica.
- Variância: Define uma medida da dispersão ou espalhamento em um conjunto de dados.
- Desvio padrão: É a raiz quadrada da variância. Conjuntos de dados com um pequeno desvio padrão são agrupados firmemente em torno da média, enquanto um desvio padrão maior indica que os dados estão mais espalhados.
- *Peak-to-Peak*: Diferença entre o valor máximo e o valor mínimo do sinal. Este atributo mede a amplitude total do sinal.
- *Crest Factor*: Definido como a razão entre os valores de pico e o valor da raiz quadrada média e indica quão extremos os picos são em uma forma de onda.
- Indicador de pulso: Indicador de pulso, calculado como o valor máximo absoluto do sinal dividido pela média dos valores do sinal. Esse atributo mede a relação entre picos e o valor médio do sinal.
- Magnitude média da aceleração: Calculada a partir dos valores de aceleração a_x , a_y e a_z . Fornece uma medida geral da magnitude da aceleração.
- Magnitude média do giroscópio: Calculada a partir dos valores de giroscópio g_x , g_y e g_z . Fornece uma medida geral da magnitude da rotação.

- Média Amostral: Cada amostra é dividida em 27 blocos de tamanho igual, em seguida é calculada a média dos valores em cada bloco. É usada para entender as características da amostra em diferentes intervalos de tempo, capturando variações e tendências ao longo da amostra.

Atributos no Domínio da Frequência

- Mediana das Magnitudes das Frequências: Mostra a distribuição das magnitudes das frequências e se há uma concentração de energia em frequências médias.
- Valor Máximo das Magnitudes das Frequências: Indica a frequência com a maior intensidade no sinal. Se essa frequência for muito alta, pode sugerir a presença de ruído ou uma característica dominante específica no sinal.
- Assimetria das Magnitudes das Frequências: Um valor de assimetria positivo indica que a distribuição tem uma cauda mais longa à direita (frequências altas têm magnitudes maiores), enquanto um valor negativo indica uma cauda mais longa à esquerda (frequências baixas têm magnitudes maiores). Isso pode indicar a predominância de componentes de frequência alta ou baixa no sinal.
- Variância das Magnitudes das Frequências: Uma alta variância indica que as magnitudes das frequências são amplamente distribuídas, sugerindo um sinal com uma variedade de componentes de frequência. Uma baixa variância indica que as magnitudes são mais concentradas em torno da média, sugerindo um sinal com componentes de frequência mais uniformes.
- Média Intervalar do Módulo da *Fast Fourier Transform* (FFT) ao Quadrado: É dividido o espectro de frequências em blocos de tamanho 27 e calculado a energia média em cada bloco. Isso revela como a energia do sinal está distribuída ao longo do espectro de frequências e identifica regiões do espectro onde a energia é mais concentrada.

Componentes de Wavelet:

- Componentes Wavelet dos Níveis 3 e 5: A decomposição *Wavelet* é uma técnica empregada para analisar sinais em diferentes escalas ou resoluções, sendo particularmente útil na identificação de características locais e globais do sinal. Neste contexto, os coeficientes do nível 3 capturam detalhes do sinal em uma escala intermediária, enquanto os coeficientes

do nível 5 capturam detalhes do sinal em uma escala mais ampla.

4.4 Classificação

O algoritmo de classificação foi executado no sistema operacional *Microsoft Windows 11*, utilizando um computador portátil *Acer Nitro AN515-45*. Este equipamento é configurado com um processador *AMD Ryzen 7 5800H* com *Radeon Graphics*, operando a 3.20 GHz, e com armazenamento SSD (*Solid State Drive*) de 500 GB. A memória RAM é de 8,00 GB e o sistema é baseado em arquitetura x64. A classificação foi realizada utilizando Python 3.12, uma das linguagens de programação mais empregadas em AM. Python é escolhido por sua sintaxe simples e pela ampla disponibilidade de bibliotecas, como NumPy, SciPy e Matplotlib (MIHAJLOVIĆ *et al.*, 2020), sendo algumas das mais utilizadas e desempenham um papel importante também neste estudo.

Os classificadores testados foram: RF, MLP, XGBOOST, SVM. A escolha desses classificadores foi fundamentada em uma revisão bibliográfica abrangente, realizada no início desse trabalho, que destacou a eficácia desses classificadores em problemas similares, além dos testes realizados. A seleção dos hiperparâmetros dos classificadores é um passo fundamental na etapa de modelagem, ao determinar a eficácia dos modelos preditivos. Para garantir uma avaliação justa e otimizada, optou-se por testar todos os classificadores com todos os atributos definidos na Seção 4.3. Este procedimento envolveu a realização de diversos testes, variando os hiperparâmetros e a quantidade de componentes de PCA em cada classificador testado. A combinação ótima de hiperparâmetros e componentes de PCA foi selecionada com base na melhor taxa de acerto obtida durante os experimentos.

O processo de ajuste e avaliação dos modelos envolveu a utilização de técnicas de validação cruzada para garantir que os resultados obtidos fossem robustos e generalizáveis. A validação cruzada permite uma estimativa mais precisa do desempenho do classificador ao dividir os dados em múltiplos subconjuntos e avaliar esse classificador em subconjuntos distintos do conjunto de dados, conforme discutido na Subseção 2.3.6.

5 RESULTADOS

Neste capítulo, serão apresentados os resultados obtidos a partir dos testes realizados com a base de dados detalhada na Seção 4.2. A Seção 5.1 abordará a seleção dos hiperparâmetros. Na Seção 5.2 são discutidos os resultados dos classificadores durante a fase dos testes. Em seguida, na Seção 5.3 é abordado o tempo de execução dos classificadores. Por fim, a Seção 5.4 realizará uma análise do melhor resultado obtido.

5.1 Seleção dos hiperparâmetros

Utilizando os atributos descritos na Seção 4.3, os classificadores RF, MLP, XGBOOST e SVM foram testados. Para determinar a quantidade ideal de componentes a ser utilizada no PCA, foram realizados diversos testes para maximizar a taxa de acerto, utilizando o menor número de componentes possível. Os resultados mostraram que a taxa de acerto mais alta foi obtida com 30 componentes de PCA.

Além disso, a otimização dos hiperparâmetros dos classificadores foi conduzida manualmente. Foram selecionadas e testadas individualmente combinações específicas dos principais hiperparâmetros, com o intuito de identificar a configuração que maximiza o desempenho do modelo. A opção por uma abordagem manual permitiu concentrar os testes em combinações pertinentes, fundamentadas em estudos anteriores e na literatura revisada durante a etapa inicial desse trabalho, em vez de explorar exaustivamente todas as possibilidades.

Embora a abordagem não tenha incluído uma ampla quantidade de combinações de hiperparâmetros, as configurações escolhidas demonstraram um desempenho satisfatório. Na Tabela 3 são mostrados os hiperparâmetros que obtiveram a melhor acurácia durante os testes.

Tabela 3 – Melhores hiperparâmetros para os classificadores

Classificador	Parâmetros utilizados
RF	$n_estimators = 150$ $max\ depth = None$ $critério = entropy$
MLP	$hidden\ layer\ sizes = 100$ $activation = relu$ $solver = adam$ $\alpha = 0.05$ $learning\ rate = adaptive$
XGBOOST	$eval\ metric = mlogloss$
SVM	$kernel\ rbf$ $one-vs-one$ $C = 10$ $\gamma = \frac{1}{n_features \cdot var(features)}$

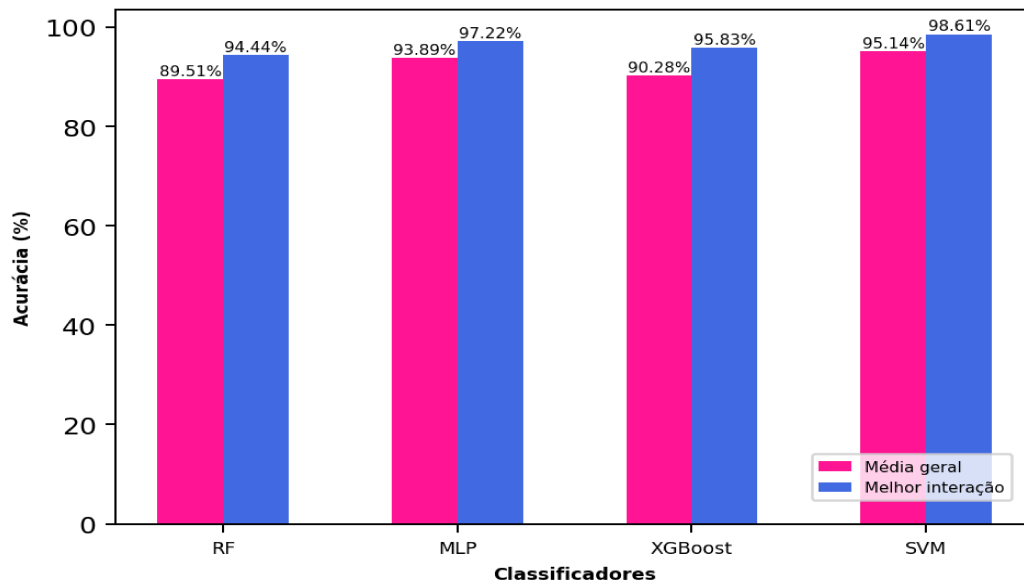
Fonte: elaborada pelo autor (2024).

5.2 Análise Geral dos Resultados da Classificação

Os testes foram conduzidos empregando a validação cruzada (*K-fold*) com $K = 20$ e os resultados obtidos demonstram a eficácia dos classificadores selecionados. Na Figura 13 são mostrados os resultados da acurácia dos classificadores, observa-se que o SVM apresentou o melhor desempenho, alcançando uma acurácia média de 95,14% e uma acurácia máxima de 98,61% na melhor interação da validação cruzada. Esses resultados indicam que o SVM se mostrou particularmente eficaz para o presente trabalho, com uma taxa de acertos satisfatória considerando o número de classes envolvidas.

Em seguida, o MLP obteve um desempenho também expressivo, com uma acurácia média de 93,88% e uma máxima de 97,22%. A rede neural MLP, apesar de sua maior complexidade, mostrou-se robusta e capaz de generalizar bem os padrões dos dados após a redução de dimensionalidade aplicada pelo PCA.

Figura 13 – Resultado dos testes utilizando 30 componentes de PCA.



Fonte: elaborada pelo autor (2024).

O XGBOOST, embora tenha obtido uma acurácia menor em comparação ao SVM e MLP, ainda assim apresentou resultados sólidos, com uma acurácia média de 90,27% e uma máxima de 95,83%. Por fim, o RF apresentou uma acurácia média de 89,51% e uma máxima de 94,44%, sendo o classificador com desempenho relativamente inferior aos demais.

5.3 Comparação dos Tempos de Execução

Para medir o tempo de execução dos classificadores, foi utilizada a biblioteca *time* do *Python*, por meio da função *time.time()*, que fornece o tempo atual em segundos desde a época, estabelecida como 1º de janeiro de 1970, 00:00:00 (UTC - *Coordinated Universal Time*).

Primeiramente, antes de iniciar a execução dos classificadores, registrou-se o tempo inicial utilizando o comando *start_time = time.time*, que armazena o número de segundos decorridos desde a época até aquele momento específico, fornecendo assim um ponto de referência para o início da medição.

Na fase seguinte, durante a execução dos classificadores, que incluiu tanto o treinamento quanto a avaliação, todas as operações computacionais, como aplicação de PCA (quando utilizada) e a validação cruzada, foram realizadas. Este processo foi essencial para garantir que o tempo de execução medido refletisse todas as etapas envolvidas durante a etapa de classificação.

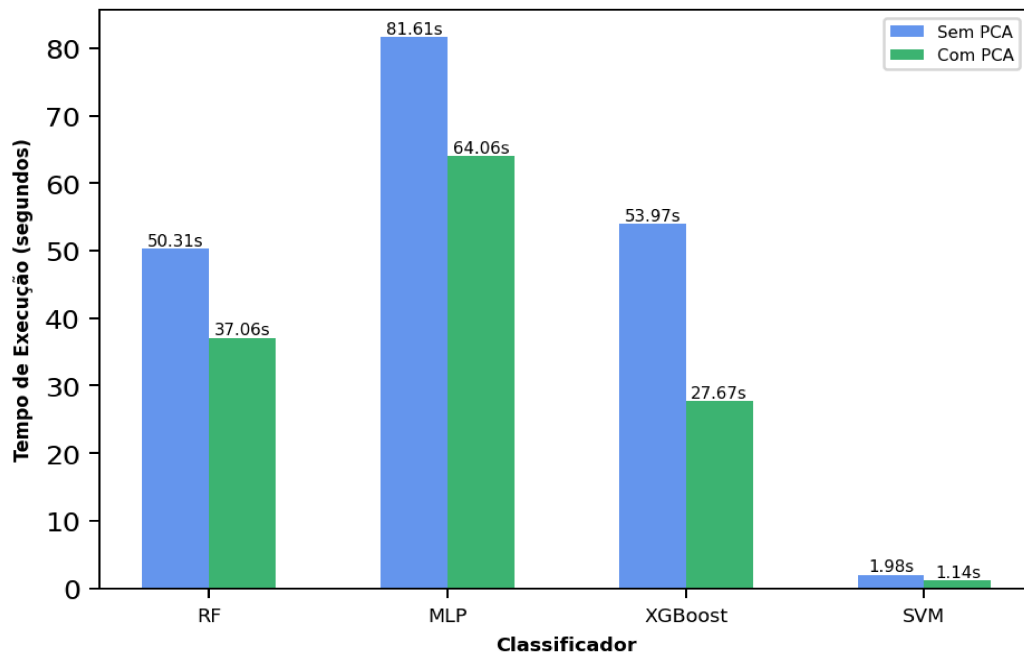
Após a conclusão dessas operações, procedeu-se ao registro do tempo final com o comando *end_time = time.time*, que capturou novamente o tempo atual em segundos desde a época. Com o tempo inicial e final devidamente registrados, a duração total foi determinada pela subtração do tempo inicial do tempo final. O valor obtido representa o tempo total gasto na execução dos classificadores, expresso em segundos

Na Figura 14 é mostrado o gráfico comparativo dos tempos de execução, nota-se que a aplicação do PCA alterou a eficiência dos classificadores. Para todos os classificadores, houve uma redução no tempo de execução após a aplicação do PCA. O XGBOOST apresentou a maior diminuição no tempo de processamento, reduzindo de 53,97 segundos sem PCA para 27,67 segundos com PCA. Esse resultado destaca a eficiência do PCA em agilizar o processo de classificação, principalmente em algoritmos que exigem um maior poder de processamento.

O MLP também se beneficiou da redução dimensional, com o tempo de execução caindo de 81,61 segundos para 64,06 segundos. Embora ainda seja o classificador com maior tempo de processamento após aplicação do PCA. O RF teve uma redução de tempo menos expressiva, passando de 50,31 segundos para 37,06 segundos.

Por fim, o SVM, que já possuía o menor tempo de execução entre os classificadores, mostrou-se ainda mais eficiente após a aplicação do PCA, reduzindo o tempo de 1,98 segundos para 1,14 segundos. Esse resultado reforça o SVM como uma excelente opção para aplicações que demandam alta acurácia e baixo tempo de processamento, características que se mostraram fundamentais para esse trabalho.

Figura 14 – Comparação dos tempos de execução dos classificadores com e sem PCA.



Fonte: elaborada pelo autor (2024).

5.4 Análise do melhor caso

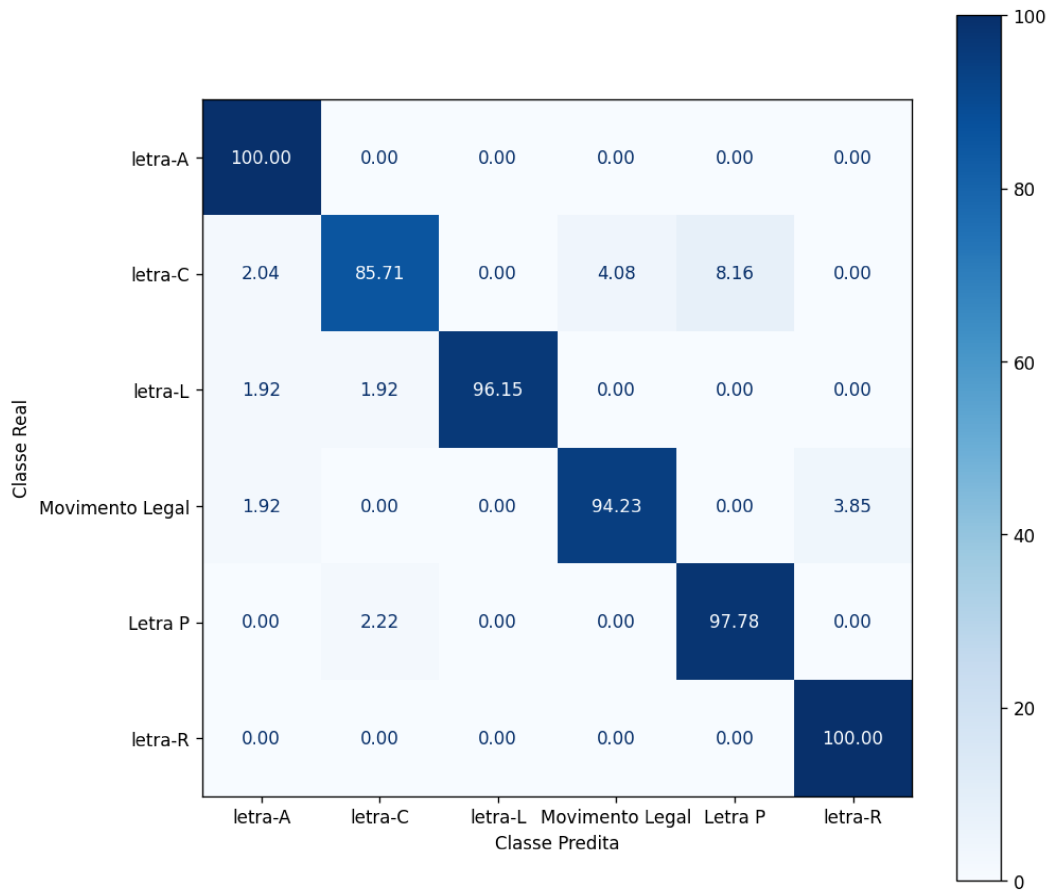
Na Figura 15 é ilustrada a matriz de confusão do classificador SVM com aplicação da PCA com 30 componentes. Essa foi a configuração que obteve o melhor desempenho durante a etapa de classificação dos movimentos de mão. A diagonal principal da matriz representa as classificações corretas para cada classe, enquanto os valores fora da diagonal indicam os erros de classificação.

Analisando a matriz de confusão, nota-se que as classes “letra A” e “letra R” foram classificadas com 100% de acerto. Isso demonstra que o SVM com PCA foi extremamente eficaz em identificar corretamente os movimentos correspondentes a essas classes, sem nenhum erro de classificação.

As classes “letra L” e “letra P” também mostraram resultados muito positivos, com uma pequena taxa de erro observada na classe “letra L”, onde 1.92% dos movimentos foram classificados como pertencentes à classe “letra C” e outros 1.92% como classe “letra P”. Na classe “letra P”, 2.22% dos movimentos foram erroneamente atribuídos à classe “letra C”.

Existem dois movimentos que obtiveram uma menor taxa de acerto. Na classe “letra C”, 2.04% dos movimentos foram erroneamente classificados como pertencentes à classe “letra A”, 4.08% foram atribuídos à classe “Movimento Legal”, e 10,2% foram classificados

Figura 15 – Matriz de confusão - SVM com PCA (%).



Fonte: elaborada pelo autor (2024).

como classe “letra P”. Essa baixa taxa de acerto pode ser atribuída a variações na execução do movimento pelo usuário, como a posição de abertura da mão, que pode ter sido feita pelo lado direito ou esquerdo, ou o ângulo de abertura na formação da letra C, que pode ter sido muito fechado.

Na classe “Movimento Legal”, 1.92% dos movimentos foram classificados incorretamente como pertencentes à classe “letra A”, e 3.85% foram atribuídos à classe “letra R”. Embora o desempenho ainda seja elevado (94.23%), esses pequenos erros sugerem que, em algumas situações, o classificador teve dificuldade em distinguir esses movimentos específicos de outros movimentos similares.

A Tabela 4 mostra as acurácias e os desvios padrão de cada acurácia, obtidos durante as 20 iterações da validação cruzada para o melhor caso. A acurácia máxima de 100% nas iterações 5, 8 e 13 demonstra a capacidade do modelo de alcançar resultados perfeitos em algumas configurações, o que é um indicativo positivo sobre a robustez do classificador.

No entanto, a variação nas acurácias individuais (de 87.50% a 100.00%) e os desvios padrão associados (que variam de 0.00% a 6.36%) sugerem que o desempenho do classificador

pode ser sensível às variações nos dados de treinamento e teste ou às dificuldades em distinguir algumas das classes.

Tabela 4 – Acurácias e desvios padrão dos classificadores em 20 iterações

Iteração	Acurácia (%)	Desvio Padrão (%)
1	91.67	4.18
2	87.50	6.36
3	87.50	6.36
4	94.44	3.02
5	100.00	0.00
6	93.06	3.80
7	98.61	2.57
8	100.00	0.00
9	95.83	2.97
10	94.44	3.02
11	93.06	3.80
12	95.83	2.97
13	100.00	0.00
14	97.22	2.36
15	97.22	2.36
16	94.44	3.02
17	98.61	2.57
18	98.61	2.57
19	93.06	3.80
20	95.83	2.97

Fonte: elaborada pelo autor (2024).

A Tabela 5 mostra as métricas de desempenho para cada classe, incluindo precisão, sensibilidade e *F1-Score*, fornecendo uma visão detalhada sobre o desempenho do classificador em relação a cada classe específica. Para o cálculo dessas métricas, foi utilizada a abordagem *one-vs-rest* (um contra o resto). Em problemas de classificação multiclasse, essas métricas são calculadas individualmente para cada classe, tratando-a como a classe positiva, enquanto as demais são agrupadas como classe negativa. As fórmulas para o cálculo dessas métricas foram definidas na Subseção 2.3.6.

Tabela 5 – Métricas de desempenho para cada classe

Classe	Precisão (%)	Sensibilidade (%)	F1-Score (%)
Letra A	95	97	96
Letra C	94	91	92
Letra L	99	97	98
Movimento Legal	96	99	98
Letra P	92	90	91
Letra R	98	99	98

Fonte: elaborada pelo autor (2024).

Analisando a Tabela 5, nota-se que as classes “letra L”, “Movimento Legal” e “letra R” alcançaram um *F1-Score* de 98%, indicando um excelente equilíbrio entre precisão e sensibilidade. Em contrapartida, a classe "letra C" apresentou um *F1-Score* de 92%, refletindo os desafios já mencionados na distinção dessa classe em comparação com as outras. No geral, esses resultados demonstram o potencial do sistema proposto para aplicações práticas de reconhecimento de gestos.

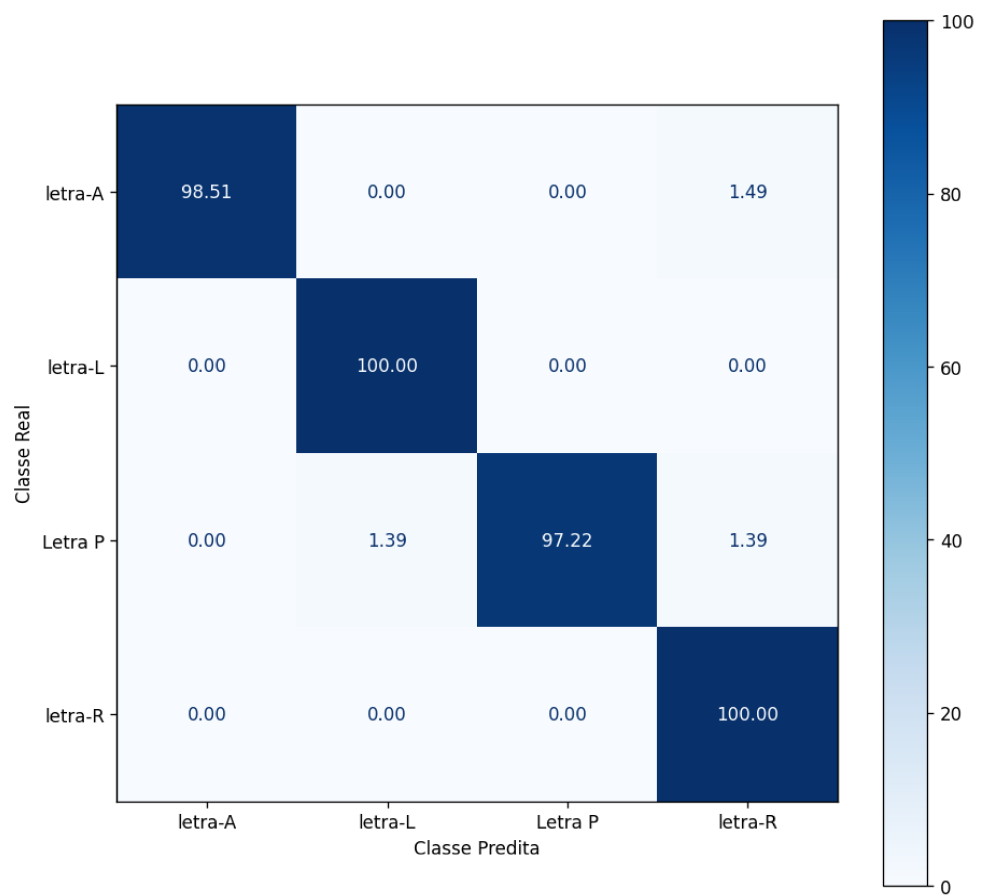
5.4.1 Análise do melhor caso com redução para 4 classes

Em certas aplicações, pode não ser necessário utilizar todas as classes, permitindo a eliminação de algumas delas para simplificação. Inicialmente, os testes foram realizados com seis classes para avaliar o desempenho geral do sistema. Em seguida, focou-se em quatro classes, removendo as duas classes com menor desempenho durante a fase de classificação. Essa abordagem possibilitou uma análise mais detalhada dos movimentos mais relevantes, otimizando o sistema e oferecendo uma alternativa para aplicações mais simples.

Neste caso ao reduzir o número de classes para quatro e realizar uma nova simulação, foi obtida a matriz de confusão para o melhor caso, apresentada na Figura 16. Nessa configuração, a taxa média de acerto atingiu 98,93%, demonstrando um aumento significativo na precisão em comparação à classificação com todas as classes.

Embora ainda haja alguns erros, como na classe “letra P”, o desempenho geral indica que o classificador se adaptou bem ao conjunto de dados reduzido. Esses resultados sugerem que a remoção de classes menos relevantes pode melhorar a acurácia do sistema para as classes restantes. A flexibilidade do modelo em ajustar-se às necessidades específicas da aplicação demonstra sua eficácia e capacidade de adaptação em contextos mais restritos.

Figura 16 – Matriz de confusão com 4 classes - SVM com PCA (%).



Fonte: elaborada pelo autor (2024).

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou o desenvolvimento de um protótipo de um sistema de reconhecimento de gestos das mãos utilizando o microcontrolador ESP32 e sensores MPU-6050, visando a utilização futura dos movimentos de mãos classificados como interface para controlar computadores.

Os resultados obtidos demonstraram que o SVM foi o classificador com o melhor desempenho geral, alcançando uma acurácia média de 95,14%. Quando analisado especificamente para quatro classes, o SVM obteve 100% de acurácia nas classes "L" e "R", 98,51% na classe "A" e 97,22% na classe "P". Além da acurácia, as métricas de precisão, sensibilidade e *F1-Score* reforçam o desempenho do modelo, mostrando uma capacidade consistente de distinguir corretamente entre as diferentes classes de gestos.

Ademais, a análise dos tempos de execução revelou que a aplicação do PCA não só melhorou a acurácia em alguns casos, como também reduziu significativamente o tempo de processamento dos classificadores. O SVM, que já possuía o menor tempo de execução entre os classificadores testados, apresentou um tempo de execução final de apenas 1.14 segundos após a aplicação do PCA.

Um aspecto significativo deste estudo foi a comparação entre os diferentes classificadores. O MLP também demonstrou um desempenho sólido, alcançando uma acurácia de 93,89%, embora com um tempo de processamento superior ao do SVM. Por outro lado, o XGBOOST e o RF, apesar de seu potencial, mostraram acurácias de 90,28% e 89,51%, respectivamente, não atingindo o mesmo nível de precisão e eficiência do SVM.

O sistema desenvolvido demonstra um alto potencial para aplicações práticas de controle de computadores por meio de gestos, apresentando uma solução inovadora para interações mais naturais e inclusivas. A combinação do ESP32 com os sensores MPU-6050 mostrou-se eficiente na captura e processamento dos sinais de movimento das mãos, validando a viabilidade do protótipo para esse propósito.

Como trabalhos futuros, propõe-se a integração dos movimentos classificados a comandos específicos em ambientes de trabalho ou aplicativos. Isso exigirá um *software* que interprete os gestos e os traduza em comandos, proporcionando uma interface mais fluida e intuitiva. A continuidade deste estudo ampliará as possibilidades de interação humano-computador, aprimorando como os usuários interagem com a tecnologia por meio de gestos naturais.

REFERÊNCIAS

- ABE, S. **Support Vector Machines for Pattern Classification**. [S. l.]: SpringerVerlang, 2005.
- AGGARWAL, C.; YU, P. S. Detecção de outlier em dados de alta dimensão. **edição regular**, 2021.
- AL-SAEDI, A. K. H.; AL-ASADI, A. H. H. Survey of hand gesture recognition systems. **Journal of Physics: Conference Series**, IOP Publishing, v. 1294, n. 4, p. 042003, sep 2019. Disponível em: <https://dx.doi.org/10.1088/1742-6596/1294/4/042003>.
- ALLENDE-CID, H. Machine learning: catalisador da ciência. **Computação Brasil**, v. 39, n. 1, p. 1–46, 2019. Disponível em: <https://www.sbc.org.br>.
- ALON, J.; ATHITSOS, V.; YUAN, Q.; SCLAROFF, S. A unified framework for gesture recognition and spatiotemporal gesture segmentation. **IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)**, September 2009. Pre-print.
- ALPAYDIN, E. **Introduction to Machine Learning**. [S. l.]: MIT Press, 2010.
- ANDRADE, E. **ESP32 DevkitC v1**. 2022. Disponível em: <https://includemicro.com/esp32-devkitcv1/>.
- ANUSHA, L.; DEVI, Y. U. Implementation of gesture based voice and language translator for dumb people. In: **2016 International Conference on Communication and Electronics Systems (ICCES)**. [S. l.: s. n.], 2016. p. 1–4.
- APONNO, D.; ARIFIN, A.; FATONI, M. H.; NUH, M.; WATANABE, T. Evaluation of multi-sensor gloves development as an assistive interpreter for indonesian sign language. In: **2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)**. [S. l.: s. n.], 2022. p. 179–184.
- BABIUCH, M.; FOLTÝNEK, P.; SMUTNÝ, P. Using the esp32 microcontroller for data processing. In: **2019 20th International Carpathian Control Conference (ICCC)**. [S. l.: s. n.], 2019. p. 1–6.
- BONATTO, A. N. **Desenvolvimento de Protótipo para Reconhecimento de Gestos de uma das Mãos**. Monografia (Trabalho de Conclusão de Curso), 2017. Apresentado à Área do Conhecimento de Ciências Exatas e Engenharias, Campus Universitário da Região dos Vinhedos.
- BREIMAN, L. Random forests. **Machine Learning**, v. 45, p. 5–32, 10 2001.
- CARNEIRO, S. B.; SANTOS, E. D. D. M.; BARBOSA, T. M. D. A.; FERREIRA, J. O.; ALCALÁ, S. G. S.; ROCHA, A. F. D. Static gestures recognition for brazilian sign language with kinect sensor. In: **2016 IEEE SENSORS**. [S. l.: s. n.], 2016. p. 1–3.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S. l.: s. n.], 2016.
- CHORZEPA, M. G.; SAEIDPOUR, A.; CHRISTIAN, J.; DURHAM, S. Hurricane vulnerability of coastal bridges using multiple environmental parameters. **International Journal of Safety and Security Engineering**, v. 6, p. 10–18, 03 2016.

CORREIA, C. H. G.; KOMATI, K. S.; BOLDT, F. d. A. Reconhecimento de gestos de mão em sequência a partir de sensores inerciais. **Journal of Health Informatics**, v. 12, mar. 2021. Disponível em: <https://jhi.sbis.org.br/index.php/jhi-sbis/article/view/841>.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, 1995. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00994018>.

DESHMUKH, A. **Microcontrollers: Theory and Applications**. McGraw-Hill Education (India) Pvt Limited, 2005. (Computer engineering series). ISBN 9780070585959. Disponível em: https://books.google.com.br/books?id=5PDx2Q9Ea_YC.

FIORIN, D. V.; MARTINS, F. R.; SCHUCH, N. J.; PEREIRA, E. B. Aplicações de redes neurais e previsões de disponibilidade de recursos energéticos solares. **Revista Brasileira de Ensino de Física**, Sociedade Brasileira de Física, v. 33, n. 1, p. 01–20, Jan 2011. ISSN 1806-1117. Disponível em: <https://doi.org/10.1590/S1806-11172011000100009>.

FRANCISCO dos S.; FLÁVIO dos S.; ALEXANDRE, C. Classification of hand movements from emg signals for people with motor disabilities. **IEEE Latin America Transactions**, v. 18, n. 11, p. 2019–2026, 2020.

GARCÍA, A. A. T.; REYES-GARCÍA, C. A.; VILLASEÑOR-PINEDA, L.; MENDOZA-MONTOYA, O. (Ed.). **Biosignal Processing and Classification Using Computational Learning and Intelligence: Principles, Algorithms, and Applications**. Elsevier Inc.: Academic Press, 2022. ISBN 978-0-12-820125-1.

GOUROB, J. H.; RAXIT, S.; HASAN, A. A robotic hand: Controlled with vision based hand gesture recognition system. In: **2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)**. [S. l.: s. n.], 2021. p. 1–4.

GUPTA, H. P. *et al.* A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. **IEEE Sensors Journal**, v. 16, n. 16, p. 6425–6432, 2016.

HSU, Y.-L.; CHU, C.-L.; TSAI, Y.-J.; WANG, J.-S. An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition. **IEEE Sensors Journal**, v. 15, n. 1, p. 154–163, 2015.

HUSSAIN, S.; SAXENA, R.; HAN, X.; KHAN, J. A.; SHIN, H. Hand gesture recognition using deep learning. In: **2017 International SoC Design Conference (ISOC)**. [S. l.: s. n.], 2017. p. 48–49.

INVENSENSE. **MPU-6000 and MPU-6050 Product Specification Revision 3.4**. Sunnyvale, 2013. Available online: <https://manuals.plus/invensense/invensense-mpu-6000-and-mpu-6050-product-specification-revision-3-4-manual-pdf>.

JAADI, Z. A step-by-step explanation of principal component analysis (pca). -, April 1 2021. Disponível em: <https://www.example.com>.

KIM, J.; MASTNIK, S.; ANDRE, E. Emg-based hand gesture recognition for realtime biosignal interfacing. In: . [S. l.: s. n.], 2008. p. 30–39.

KIM, S. Y.; HAN, H. G.; KIM, J. W.; LEE, S.; KIM, T. W. A hand gesture recognition sensor using reflected impulses. **IEEE Sensors Journal**, v. 17, n. 10, p. 2975–2976, 2017.

KUMAR, P.; RAUTARAY, S. S.; AGRAWAL, A. Hand data glove: A new generation real-time mouse for human-computer interaction. In: **2012 1st International Conference on Recent Advances in Information Technology (RAIT)**. [S. l.: s. n.], 2012. p. 750–755.

KUMAR, S. S.; GATTI, R.; KUMAR, S. K. N.; NATARAJA, N.; PRASAD, R. P.; SARALA, T. Glove based deaf-dumb sign language interpreter. In: **2021 International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT)**. [S. l.: s. n.], 2021. p. 947–950.

LEE-FERNG, J.; SOLAR, J. Ruiz-del; VERSCHAE, R.; CORREA, M. Dynamic gesture recognition for human robot interaction. In: **2009 6th Latin American Robotics Symposium (LARS 2009)**. [S. l.: s. n.], 2009. p. 1–8.

LUDERMIR, T. B. Title of the article. **Name of the Journal**, Volume Number, n. Issue Number, p. Page Range, 2021. Received on Tue, 16 Feb 2021; Accepted on Thu, 18 Feb 2021; Published on Thu, 01 Apr 2021. This is an open-access article under a Creative Commons license. Disponível em: <https://doi.org/10.1590/s0103-4014.2021.35101.007>.

MAHADEVKAR, S. V.; KHEMANI, B.; PATIL, S.; KOTECHA, K.; VORA, D. R.; ABRAHAM, A.; GABRALLA, L. A. A review on machine learning styles in computer vision—techniques and future directions. **IEEE Access**, v. 10, p. 107293–107329, 2022.

MAIER, A.; SHARP, A.; VAGAPOV, Y. Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things. In: **2017 Internet Technologies and Applications (ITA)**. [S. l.: s. n.], 2017. p. 143–148.

MARQUES, J. **Reconhecimento de Padrões Métodos Estatísticos e Neurais**. [S. l.]: IST Press, 1999.

MEYER-BÄSE, A. Ii - feature selection and extraction. In: MEYER-BÄSE, A. (Ed.). **Pattern Recognition in Medical Imaging**. San Diego: Academic Press, 2004. p. 14–49. ISBN 978-0-12-493290-6. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780124932906500055>.

MIHAJLOVIĆ, S.; KUPUSINAC, A.; IVETIĆ, D.; BERKOVIĆ, I. The use of python in the field of artificial intelligence. In: TECHNICAL FACULTY “MIHAJLO PUPIN”, ZRENJANIN, REPUBLIC OF SERBIA AND FACULTY OF TECHNICAL SCIENCES, NOVI SAD, REPUBLIC OF SERBIA. **International Conference on Information Technology and Development of Education – ITRO 2020**. Zrenjanin, Republic of Serbia, 2020.

MISRA, P.; MMWAELI. Hsv brightness factor matching for gesture recognition system. In: . [S. n.], 2010. Disponível em: <https://api.semanticscholar.org/CorpusID:207914218>.

MITRA, S.; ACHARYA, T. Gesture recognition: A survey. **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews**, IEEE, v. 37, n. 3, p. 311–324, 2007.

NAVARRO, J. C.; VARGAS, F. L. P. J. B. Emg-based system for basic hand movement recognition. **DYNA**, SCIELO, v. 79, n. 171, 2010.

PATIL, M.; DISALE, S.; PHALKE, H.; RANADE, U.; KHOT, P. S.; ENGINEERING, C.; PATIL, D. D. Y. Survey on gesture based virtual mouse and keyboard. In: . [S. n.], 2022. Disponível em: <https://api.semanticscholar.org/CorpusID:249060331>.

PEREIRA, G. M. S. **Mapeamento do uso e cobertura da terra utilizando séries temporais de imagens do satélite VENUS e algoritmos de aprendizado de máquina.** Dissertação (Dissertação de Mestrado) – Universidade de Campinas, Faculdade de Engenharia Agrícola, Campinas, 2021.

PRADO, F. J. J. **Engenharia de Computação e Tecnologias Assistivas: Recursos de Acessibilidade Ao Computador para Pessoas Com Deficiência Motora.** Dissertação (Mestrado em Engenharia de Computação) – Universidade Federal do Ceará - Campus Sobral, Sobral, 2020.

QUIRINO, F. A.; ROMANSSINI, M.; BALDISSERA, M.; GIRARDI, A. G. System implementation of a gesture detection glove for human-computer interaction. In: -. [S. n.], 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:202637838>.

REDDY, V. V.; DHYANCHAND, T.; KRISHNA, G. V.; MAHESHWARAM, S. Virtual mouse control using colored finger tips and hand gesture recognition. In: **2020 IEEE-HYDCON**. [S. l.: s. n.], 2020. p. 1–5.

RINGNER, M. What is principal component analysis? **Nature Biotechnology**, v. 26, n. 3, p. 303–304, March 2008. ISSN 1546-1696. Disponível em: <https://doi.org/10.1038/nbt0308-303>.

RODRIGUES, V. Métricas de avaliação: acurácia, precisão, recall. . . quais as diferenças? April 12 2019.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. [S.l.]: Prentice Hall, 2002.

RYLO, M. N. **Reconhecimento de gestos de membros superiores utilizando sensores de movimento e fotopletagemografia.** Dissertação (Dissertação de Mestrado) – Universidade Federal do Amazonas - UFAM, Manaus, AM, Abril 2023.

SAMAT, A.; ZHAO, Y.; DU, P.; LIU, S.; GHAMISI, P.; KANG, J. Meta-xgboost for hyperspectral image classification using extended mser-guided morphological profiles. **Remote Sensing**, MDPI, v. 12, n. 12, p. 1973, 2020.

SANTOS, F. V. dos. **Sistema de Aquisição e Classificação de Movimentos de Mãos Utilizando Acelerômetros, Giroscópios e Reconhecimento de Padrões.** Monografia (Trabalho de Conclusão de Curso) – Universidade Federal do Ceará, Campus Sobral, Sobral, 2020.

SANTOS, F. V. dos. **Funções Kernel tensoriais baseadas em tensores-núcleo aplicadas à classificação de movimento de mãos.** Dissertação (Mestrado em Engenharia Elétrica e de Computação) – Universidade Federal do Ceará, Campus de Sobral, 2023. Disponível em: <http://repositorio.ufc.br/handle/riufc/76230>.

SHARMA, S.; JAIN, S.; KHUSHBOO. A static hand gesture and face recognition system for blind people. In: **2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)**. [S. l.: s. n.], 2019. p. 534–539.

SILVA, F. T. **Aplicação do Algoritmo Ensemble de Floresta Aleatória para a Classificação de Clientes Adimplentes e Inadimplentes.** Dissertação (Trabalho de Conclusão de Curso) – Universidade Federal de Mato Grosso, Campus Universitário de Várzea Grande, Faculdade de Engenharia, Cuiabá, MT, maio 2021. Curso de Engenharia de Computação da FAENG/CUVG/UFMT (área de concentração: Engenharia de Computação).

M. F. Tappen *et al.* **Cluster-trained machine learning for image processing**. 2017. US Patent 9,704,054.

THANTILAGE, R. D.; LE-KHAC, N.-A.; KECHADI, M.-T. Healthcare data security and privacy in data warehouse architectures. **Informatics in Medicine Unlocked**, v. 39, p. 101270, 2023. ISSN 2352-9148. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352914823001144>.

THEODORIDIS, S.; KOUTROUBAS, K. Feature generation ii. In: **Pattern Recognition**. 4th. ed. [S. l.]: Academic Press, 2009. cap. 7, p. 411–479.

TOU, J. **Computer-Based Particle Shape Analysis for Classification, Recognition, Utilization**. [S. l.]: Advanced Particulate Morphology, 2018. ISBN 9781351069540.

WANG, H.; ZHANG, H. Movie genre preference prediction using machine learning for customer-based information. In: **2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)**. [S. l.]: IEEE, 2018. p. 110–116. Citado na página 14.

WANG, Y.; PAN, Z.; ZHENG, J.; QIAN, L.; MINGTAO, L. A hybrid ensemble method for pulsar candidate classification. **Astrophysics and Space Science**, v. 364, 08 2019.

XGBOOST. **XGBoost Documentation**. [S. l.], 2022. Acesso em junho de 2024. Disponível em: <https://xgboost.readthedocs.io/en/stable/>.

XIE, R.; CAO, J. Accelerometer-based hand gesture recognition by neural network and similarity matching. **IEEE Sensors Journal**, v. 16, n. 11, p. 4537–4545, 2016.

ZHANG, X.; CHEN, X.; LI, Y.; LANTZ, V.; WANG, K.; YANG, J. A framework for hand gesture recognition based on accelerometer and emg sensors. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans**, v. 41, n. 6, p. 1064–1076, 2011.