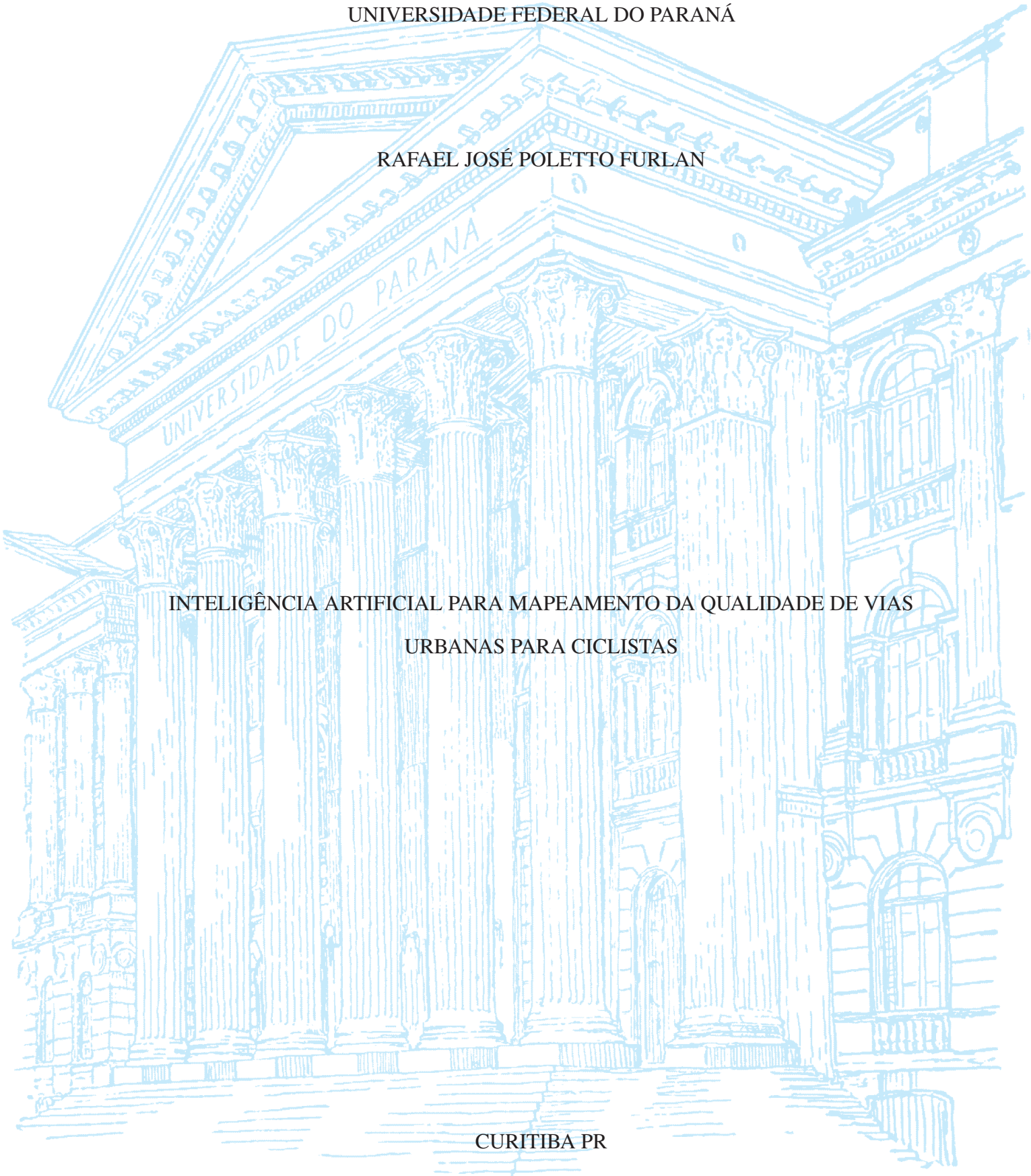
UNIVERSIDADE FEDERAL DO PARANÁ

RAFAEL JOSÉ POLETTO FURLAN



INTELIGÊNCIA ARTIFICIAL PARA MAPEAMENTO DA QUALIDADE DE VIAS URBANAS PARA CICLISTAS

CURITIBA PR

2025

RAFAEL JOSÉ POLETTO FURLAN

INTELIGÊNCIA ARTIFICIAL PARA MAPEAMENTO DA QUALIDADE DE VIAS URBANAS PARA CICLISTAS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Computação*.

Orientador: Luciano Silva.

CURITIBA PR

2025

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Furlan, Rafael José Poletto
Inteligência artificial para mapeamento da qualidade de vias urbanas para ciclistas / Rafael José Poletto Furlan. – Curitiba, 2025.
1 recurso on-line : PDF.

Dissertação (Dissertação) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: Luciano Silva

1. Aprendizado do computador. 2. Ciclovias – Segurança. 3. Planejamento Urbano. 4. Inteligência artificial. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Informática. III. Silva, Luciano. IV. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **RAFAEL JOSÉ POLETTO FURLAN**, intitulada: **INTELIGÊNCIA ATIFICIAL PARA MAPEAMENTO DA QUALIDADE DE VIAS URBANAS PARA CICLISTAS**, sob orientação do Prof. Dr. LUCIANO SILVA, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 27 de Março de 2025.

Assinatura Eletrônica
01/04/2025 16:55:33.0
LUCIANO SILVA
Presidente da Banca Examinadora

Assinatura Eletrônica
01/04/2025 14:48:14.0
GARRONE RECK
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
31/03/2025 17:09:29.0
EDUARDO CUNHA DE ALMEIDA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Documento assinado eletronicamente de acordo com o disposto na legislação federal Decreto 8539 de 08 de outubro de 2015.
Gerado e autenticado pelo SIGA-UFPR, com a seguinte identificação única: 437746
Para autenticar este documento/assinatura, acesse https://siga.ufpr.br/siga/visitante/autenticacaoassinaturas.jsp e insira o codigo 437746

RESUMO

Este estudo acadêmico explora o uso de aprendizado de máquina, com o objetivo de desenvolver um algoritimo para coleta e análise dados de sensores de dispositivos móveis para mapear as condições das vias urbanas no ciclismo. O objetivo desta pesquisa visa melhorar a experiência e a segurança dos ciclistas, além de oferecer dados atualizados para entidades públicas, auxiliando no planejamento e manutenção da infraestrutura viária. A pesquisa envolve a investigação de sensores como: acelerômetro, giroscópio, GPS e câmeras; e um aplicativo que integra informações destes dispositivos utilizando técnicas de aprendizado de máquina em uma interface intuitiva para dispositivos móveis. A metodologia contou com o levantamento bibliográfico e discussão dos principais trabalhos nesta temática, incluindo aplicações em áreas correlatas, além da própria coleta de 453.951 amostras de dados dos sensores, adquiridos durante diversos percursos em vias urbanas da cidade de Curitiba, o pré-processamento e a rotulação supervisionada destes dados, a definição e treinamento de diversos modelos de aprendizado de máquina e validação preliminar desta abordagem. Os experimentos foram realizados em diversas condições de vias públicas e com diferentes arquiteturas de coleta de dados, de dispositivos móveis acoplados em variadas posições, utilizando 2 modelos de biciletas populares. Após a análise qualitativa e quantitativa desses experimentos, pode-se concluir que o estudo identificou o modelo de Florestas Aleatórias como o mais efeito para a aplicação final, alcançando um F1-score de 0,999. Redes Neurais também apresentaram desempenho significativo, com um F1-score de 0,990, após extensivo ajuste de hiperparâmetros e seleção de variáveis. Os resultados demonstram a viabilidade da utilização de sensores de dispositivos móveis e técnicas de aprendizado de máquina para identificação automática das condições de vias urbanas durante a atividade do ciclismo. Além disso, o desenvolvimento do aplicativo permitiu a coleta eficiente de dados e o desenvolvimento de um sistema computacional consistente para futuras ações de planejamento urbano. Este trabalho também abre caminhos para pesquisas futuras, como por exemplo: a ampliação e compartilhamento de dados das vias públicas, a integração com plataformas de mapeamento e a exploração de técnicas de aprendizado contínuo, visando aprimorar ainda mais a segurança e a experiência dos ciclistas. O estudo também estabelece um ponto de partida para futuras pesquisas que explorem novas tecnologias e algoritmos, contribuindo para a evolução das soluções de mobilidade urbana sustentável.

Palavras-chave: Aprendizado de Máquina; Sensores de Dispositivos Móveis; Mapeamento de Vias Urbanas; Segurança no Ciclismo; Planejamento Urbano

ABSTRACT

This academic study explores the use of machine learning to develop an algorithm for collecting and analyzing data from mobile device sensors to map urban road conditions in cycling. The objective of this research is to enhance cyclists' experience and safety while providing updated data to public entities, aiding in road infrastructure planning and maintenance. The study investigates sensors such as accelerometers, gyroscopes, GPS, and cameras, along with an application that integrates information from these devices using machine learning techniques in an intuitive mobile interface. The methodology included a literature review and discussion of key works in this field, including applications in related areas, as well as the collection of 453,951 sensor data samples acquired during various cycling routes in urban streets of Curitiba. The research also involved preprocessing and supervised labeling of the data, defining and training multiple machine learning models, and conducting a preliminary validation of this approach. Experiments were conducted under various road conditions and with different data collection architectures, using mobile devices mounted in different positions on two popular bicycle models. After a qualitative and quantitative analysis of these experiments, the study identified the Random Forest model as the most effective for the final application, achieving an F1-score of 0.999. Neural Networks also demonstrated significant performance, reaching an F1-score of 0.990 after extensive hyperparameter tuning and feature selection. The results demonstrate the feasibility of using mobile device sensors and machine learning techniques for the automatic identification of urban road conditions during cycling activities. Additionally, the development of the application enabled efficient data collection and the creation of a robust computational system for future urban planning actions. This work also paves the way for future research, such as expanding and sharing public road data, integrating with mapping platforms, and exploring continuous learning techniques to further improve cyclists' safety and experience. Moreover, the study establishes a foundation for future research on emerging technologies and algorithms, contributing to the advancement of sustainable urban mobility solutions.

Keywords: Machine Learning; Mobile Device Sensors; Urban Road Mapping; Cycling Safety; Urban Planning

LISTA DE FIGURAS

2.1 Tipos de aprendizado de máquina 12

4.1 Tela inicial do aplicativo com três botões principais: Executar, Mais Informações e Configurações. Executar, leva o usuário a tela de coleta de dados, Informações, conduz o usuário a uma tela com mais dados do aplicativo, Configurações, permite o usuário habilitar e desabilitar sensores.. 22

4.2 Tela das configurações dos sistema proposto, é essencial para garantir flexibilidade na coleta de dados, permitindo que o usuário habilite ou desabilite sensores específicos conforme necessário. 22

4.3 Tela de captura de dados - representa a interface do aplicativo desenvolvido para a coleta de informações durante os percursos realizados pelos ciclistas. Esta tela permite que o usuário visualize em tempo real as informações provenientes de cada sensor do dispositivo móvel utilizado. Os principais elementos observados incluem: Visualização em tempo real dos dados dos sensores: O usuário pode acompanhar as leituras instantâneas de todos os sensores ativados durante o percurso, possibilitando um monitoramento direto das condições da via. Configuração do intervalo de captura: Há a possibilidade de definir a frequência com que os dados são registrados, permitindo ajustar a granularidade da coleta conforme necessário. Definição da rotulação: A interface oferece a opção de atribuir rótulos aos dados coletados, facilitando a posterior categorização e análise das informações. Escolha da pasta de armazenamento: O usuário pode determinar onde os dados serão salvos no dispositivo, proporcionando organização e controle sobre o gerenciamento dos arquivos gerados. Esta figura ilustra a funcionalidade principal do aplicativo desenvolvido para a pesquisa, demonstrando como os dados brutos das condições das vias urbanas eram coletados diretamente durante os trajetos de bicicleta. Os dados capturados por esta interface serviram como base para todo o processamento e análise posterior, culminando no desenvolvimento do modelo 23

4.4 Comparação de modelos de Aprendizado de Máquina - Os modelos de Redes Neurais, árvore de decisão, Gradient Boosting Classifier e Random Forest apresentam resultados significativos, com precisão, acurácia, recall e F1 score superior a 96%, entre os 4 modelos o modelo de Random Forest é o que mais se destaca nos 4 indicadores.. 25

4.5 Desempenho do modelo de Redes Neurais - mostra o desempenho do modelo em 500 épocas. As linhas azul e verde representam a precisão para treino e validação, ambas próximas de 1, indicando alta acurácia. A perda de treino (laranja) é baixa e estável, enquanto a perda de validação (vermelho) apresenta oscilações. A estabilidade da precisão indica bom aprendizado geral. 25

LISTA DE TABELAS

4.1 Métricas de desempenho dos modelos - A Random Forest obteve o melhor resultado geral, com todas as métricas próximas de 1. Árvores de Decisão e Redes Neurais também mostraram ótimo desempenho. Já o Gaussian Naive Bayes e a Regressão Logística tiveram resultados inferiores em todas as métricas. O Gradient Boosting Classifier apresentou um equilíbrio, com desempenho próximo das melhores técnicas. Os modelos mais complexos tiveram maior precisão e recall. 24

SUMÁRIO

1	**INTRODUÇÃO**	**8**
2	**REVISÃO DE LITERATURA**	**9**
2.1	SENSORES EM DISPOSITIVOS MÓVEIS	9
2.2	APRENDIZADO DE MÁQUINA	9
2.3	CONCEITOS DE APRENDIZADO DE MÁQUINA	10
2.4	CATEGORIZAÇÕES DE APRENDIZADO DE MÁQUINA	11
2.5	ALGORITMOS DE APRENDIZADO DE MÁQUINA	12
2.6	APRENDIZADO PROFUNDO E REDES NEURAIS RECORRENTES	14
2.7	APLICATIVOS EM DISPOSITIVOS MÓVEIS	14
2.8	ANÁLISE EXPLORATÓRIA DE DADOS	15
2.9	DESENVOLVIMENTO DO MODELO DE APRENDIZADO DE MÁQUINA	15
2.10	TRABALHOS RELACIONADOS	16
3	**METODOLOGIA**	**18**
3.1	OBJETIVO GERAL	18
3.2	OBJETIVOS ESPECÍFICOS	18
3.3	DESENVOLVIMENTO DO APLICATIVO	19
3.4	CONSIDERAÇÕES FINAIS SOBRE A METODOLOGIA	20
4	**RESULTADOS.**	**21**
4.1	COLETA DOS DADOS	21
4.2	PREPROCESSAMENTO DOS DADOS	21
4.3	ROTULAÇÃO DOS DADOS	21
4.4	ANÁLISE E NORMALIZAÇÃO	21
4.5	COMPARAÇÃO DOS MODELOS CLASSIFICATÓRIO E CLASSIFICAÇÃO COM CONJUNTO DE DADOS DE TESTE	24
4.6	AVALIAÇÃO DOS MODELOS CLASSIFICATÓRIOS	26
4.7	TREINAMENTO DO MODELO DE REDES NEURAIS	26
4.8	ESCOLHA DE UM MODELO	28
5	**TRABALHOS FUTUROS**	**29**
6	**CONCLUSÃO**	**30**
	REFERÊNCIAS	**32**

1 INTRODUÇÃO

Nos últimos anos, a interseção entre tecnologia móvel e aprendizado de máquina tem proporcionado avanços significativos em diversos campos, e a área de atividades físicas e esportes não é exceção. Com o crescente interesse na saúde e no bem-estar, a integração de dispositivos móveis, como smartphones, com algoritmos de aprendizado de máquina tem aberto um vasto leque de oportunidades para melhorar a experiência dos praticantes de esportes e atividades ao ar livre. Um desses esportes que se beneficia amplamente dessas inovações é o ciclismo. Segundo (Lane et al., 2010), os dispositivos móveis têm um grande potencial para revolucionar áreas como transporte e monitoramento ambiental, sendo ferramentas essenciais para a coleta de dados em tempo real.

Neste trabalho, exploramos como os sensores presentes nos dispositivos móveis podem ser utilizados em conjunto com algoritmos de aprendizado de máquina para criar um aplicativo projetado especialmente para ciclistas. O cerne desta pesquisa consiste em desenvolver um modelo capaz de coletar, processar e analisar dados provenientes dos sensores de um celular, com o objetivo de oferecer aos ciclistas orientações geográficas baseadas na infraestrutura viária e aos entes públicos informações atualizadas sobre tais condições das vias. Conforme destacado por (Fazeen et al., 2012), a utilização de sensores móveis para monitoramento de condições de estradas já demonstrou resultados promissores, com precisão superior a 85% na detecção de irregularidades.

Este trabalho se dividiu em várias etapas fundamentais: primeiro, examinamos a variedade de sensores presentes em dispositivos móveis e suas capacidades individuais. Em seguida, abordamos a aplicação de algoritmos de aprendizado de máquina, incluindo técnicas de aprendizado supervisionado e não supervisionado, para processar esses dados e criar um modelo classificatório robusto. Por fim, avaliaremos o modelo final desenvolvido. A escolha de técnicas de aprendizado profundo, como redes neurais recorrentes (RNNs), foi baseada em sua capacidade de lidar com dados sequenciais e complexos, conforme descrito por (Medsker e Jain, 2001a).

O presente trabalho teve como objetivo geral desenvolver um modelo de aprendizado de máquina classificatório, com o intuito de prever a qualidade das vias através dos sensores do celular. Entre os objetivos específicos, destacam-se:

- Desenvolver um aplicativo para a coleta de dados provenientes de sensores presentes em dispositivos móveis, tais como acelerômetro, giroscópio, GPS e câmera, com o intuito de criar uma base de dados robusta e de fonte aberta;
- Analisar os dados coletados pelo aplicativo, explorando técnicas de processamento e interpretação, identificando padrões relevantes e obtendo insights para o desenvolvimento do modelo classificatório;
- Implementar um modelo de aprendizado de máquina capaz de processar os dados dos sensores, visando a criação de um sistema classificatório eficaz na identificação da qualidade da infraestrutura viária.

Os resultados obtidos indicam que é possível identificar eventos durante o ciclismo com uma precisão de 99,9%, utilizando algoritmos como Florestas Aleatórias. Esse desempenho demonstra a viabilidade da aplicação de sensores móveis e técnicas de aprendizado de máquina para melhorar a segurança e a experiência dos ciclistas, além de fornecer dados valiosos para o planejamento de infraestrutura urbana.

2 REVISÃO DE LITERATURA

Por meio da revisão de literatura, exploramos a interseção entre sensores presentes em dispositivos móveis e algoritmos de Aprendizado de Máquina, a fim de desenvolver o aplicativo proposto pelo presente trabalho. Os sensores, como acelerômetro, giroscópio, GPS, câmera e microfone, desempenham um papel fundamental na captura de informações do ambiente em tempo real, criando assim uma base de dados que alimenta constantemente o modelo de Aprendizado de Máquina. Em paralelo, destacamos a importância do Aprendizado de Máquina na interpretação das informações sensoriais, transformando-as em informações sonoras e visuais perceptíveis para auxiliar os usuários do aplicativo. Esses algoritmos capacitam o aplicativo a identificar padrões, realizar previsões e tomar decisões com base nos dados coletados, proporcionando uma funcionalidade preditiva e adaptável crucial para aprimorar a qualidade de vida dos ciclistas.

2.1 SENSORES EM DISPOSITIVOS MÓVEIS

Os dispositivos móveis modernos são equipados com uma ampla variedade de sensores que capturam informações do ambiente ao redor do usuário, desempenhando um papel crucial em várias aplicações, desde navegação por GPS até reconhecimento de voz e monitoramento de saúde. Nesta seção, exploraremos os diferentes tipos de sensores presentes em dispositivos móveis e detalharemos as informações que cada um deles é capaz de capturar, dado que eles representam a maior parte do mercado global. Segundo dados da (StatCounter, 2023), o sistema operacional Android detinha uma expressiva fatia de 70,48% do total de dispositivos móveis em uso.

Segundo o (Google, 2023), os sensores disponíveis em dispositivos móveis modernos nas últimas versões do Android são: Acelerômetro, Giroscópio, GPS, Câmera, Microfone, Sensor de Luz Ambiente, Sensor de Proximidade, Sensor de Pressão Barométrica e Sensor de Temperatura e Umidade.

A variedade de sensores disponíveis em dispositivos móveis modernos oferece um vasto potencial para a coleta de dados em tempo real. No entanto, a heterogeneidade na disponibilidade desses sensores em diferentes dispositivos pode representar um desafio para a generalização do modelo proposto. Portanto, é essencial considerar essa variabilidade durante o desenvolvimento do aplicativo e do modelo de aprendizado de máquina.

2.2 APRENDIZADO DE MÁQUINA

Aprendizado de máquina (Aprendizado de Máquina) é um subcampo da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos estatísticos que permitem aos sistemas computacionais aprender e melhorar seu desempenho em tarefas específicas a partir da análise de dados, sem a necessidade de se realizar uma programação explícita.

De acordo com (Shinde e Shah, 2018), o aprendizado de máquina vem apresentando resultados significativos em uma variedade de domínios. Em visão computacional, técnicas de aprendizado de máquina são utilizadas para reconhecimento de objetos, detecção de padrões e segmentação de imagens. Na área de processamento de linguagem natural, o aprendizado de máquina é empregado em tarefas como análise semântica, tradução automática e geração de resumos. Além disso, em aplicações de previsão, o aprendizado de máquina é utilizado para análise de séries temporais, previsão de demanda e modelagem preditiva. Em resumo,

as aplicações do aprendizado de máquina abrangem desde a análise de imagens médicas até a detecção de fraudes em transações financeiras, demonstrando sua versatilidade e relevância em diversos contextos.

No contexto do presente trabalho, a aplicação de algoritmos de Aprendizado de Máquina desempenha um papel fundamental na extração de informações úteis dos dados capturados pelos sensores de dispositivos móveis. Isso permite que esses dispositivos automaticamente identifiquem padrões, façam previsões e tomem decisões com base nos dados sensoriais coletados e, dessa forma, torna-se evidente a importância dessa ferramenta para o desenvolvimento do trabalho proposto.

2.3 CONCEITOS DE APRENDIZADO DE MÁQUINA

Na análise de modelos de aprendizado de máquina, diversos conceitos e métricas são essenciais para avaliar o desempenho e a eficácia das predições. O **feedback** refere-se ao processo de utilizar informações geradas pelo sistema, como erros de predição, para ajustar e melhorar o desempenho do modelo. Esse mecanismo é amplamente utilizado em técnicas como o aprendizado por reforço, onde o agente recebe recompensas ou penalidades com base em suas ações, permitindo que ele aprenda a otimizar seu comportamento ao longo do tempo. Já o **overfitting** ocorre quando um modelo se ajusta excessivamente aos dados de treinamento, capturando ruídos e detalhes irrelevantes, o que prejudica sua capacidade de generalizar para novos dados. Para mitigar esse problema, técnicas como regularização, validação cruzada e dropout são frequentemente empregadas (Goodfellow et al., 2016). A **recorrência** é um conceito central em redes neurais recorrentes (RNNs), onde conexões entre neurônios formam ciclos, permitindo que o modelo capture dependências temporais em sequências de dados (Medsker e Jain, 2001b).

Além desses conceitos, métricas como **acuracia**, **precisão**, **recall** e **F1-score** são fundamentais para avaliar a qualidade dos modelos. A **acuracia** é a proporção de predições corretas sobre o total de predições realizadas, sendo amplamente usada por sua simplicidade (Provost e Fawcett, 2013). A **precisão** é definida como a proporção de exemplos classificados positivamente pelo modelo que realmente pertencem à classe positiva. Essa métrica é especialmente relevante em problemas onde o objetivo principal é minimizar a quantidade de falsos positivos, como discutido no contexto de classificação de documentos e textos (Sokolova e Lapalme, 2009). O **recall** (ou sensibilidade) mede a proporção de casos positivos reais que foram corretamente identificados pelo modelo, sendo importante em situações onde falsos negativos são inaceitáveis (Powers, 2011). Já o **F1-score** é uma métrica que combina precisão e recall em uma única medida, permitindo avaliar o desempenho de um sistema de forma balanceada, especialmente em situações onde é necessário considerar tanto os casos corretamente identificados quanto os casos perdidos (Chinchor, 1992). Outro conceito crucial é o **learning rate** (taxa de aprendizado), que controla a magnitude dos ajustes nos pesos do modelo durante o treinamento. Um valor muito alto pode levar a instabilidade, enquanto um valor muito baixo pode tornar o treinamento lento ou estagnado. A escolha adequada do learning rate é essencial para garantir a convergência eficiente do modelo (Ruder, 2016). Esses conceitos e métricas, quando bem compreendidos e aplicados, são cruciais para o desenvolvimento e avaliação de modelos de aprendizado de máquina robustos e eficazes.

2.4 CATEGORIZAÇÕES DE APRENDIZADO DE MÁQUINA

Com base em nosso estudo, exploraremos as abordagens e algoritmos de Aprendizado de Máquina mais relevantes atualmente, com o objetivo de identificar, posteriormente, a abordagem mais apropriada. Para uma compreensão abrangente do campo do Aprendizado de Máquina, é fundamental primeiro familiarizar-nos com suas categorias principais, as quais representam grupos conceituais que classificam diferentes abordagens com base em princípios gerais. Posteriormente, examinaremos os algoritmos específicos, suas características e objetivos. As principais categorias de Aprendizado de Máquina incluem:

1. **Aprendizado Supervisionado (Supervised Learning)**: O aprendizado supervisionado, conforme (Oladipupo, 2010), é uma abordagem comum em problemas de classificação, onde o objetivo é fazer com que o computador aprenda um sistema de classificação que foi criado. Ele utiliza de um conjunto de dados rotulados para treinar um modelo, no qual cada entrada é associada a uma saída desejada, de modo que o modelo treinado possa ser utilizado para fazer previsões ou classificações a partir de novos dados.

2. **Aprendizado Não Supervisionado (Unsupervised Learning)**: O aprendizado não supervisionado lida com dados não rotulados e é encarregado de encontrar padrões ou estruturas intrínsecas nesses dados. O agrupamento e a redução de dimensionalidade são exemplos comuns de aplicação do aprendizado não supervisionado, de acordo com (Ghahramani, 2003).

3. **Aprendizado por Reforço (Reinforcement Learning)**: Segundo (François-Lavet et al., 2018) o Aprendizado por Reforço é uma abordagem de aprendizado de máquina que se concentra em decisões sequenciais em ambientes incertos, inspirada pela psicologia comportamental. Nesse framework, um agente interage com o ambiente, recebendo informações sobre seu estado e respondendo com ações que são recompensadas de acordo com sua eficácia, possuindo aplicações em diversas áreas, como robótica, saúde e finanças, oferecendo uma solução flexível para otimizar objetivos em ambientes incertos através da acumulação de recompensas.

4. **Aprendizado Semi-Supervisionado (Semi-Supervised Learning)**: (Chapelle et al., 2006) define o aprendizado semi-supervisionado como uma abordagem de aprendizado de máquina que utiliza tanto dados rotulados quanto não rotulados para treinar modelos, visando aprimorar o desempenho em comparação com o aprendizado supervisionado. Em contextos como a classificação de documentos, onde apenas alguns são rotulados, ou no reconhecimento de padrões em imagens médicas, em que a maioria das imagens é não rotulada, o aprendizado semi-supervisionado se destaca ao incorporar informações dos dados não rotulados, melhorando a eficácia e generalização dos modelos de aprendizado de máquina.

5. **Aprendizado por Transferência (Transfer Learning)**: Transferência de aprendizado, de acordo com (Weiss et al., 2016), é uma técnica de aprendizado de máquina que permite que um modelo treinado em uma tarefa seja reutilizado em outra tarefa relacionada, sendo útil em situações em que há poucos dados disponíveis para treinar um modelo em uma tarefa específica. Existem várias abordagens para a transferência de aprendizado, incluindo a transferência de recursos, a transferência de conhecimento e a transferência de aprendizado por meio de adaptação de domínio. A transferência de aprendizado tem sido amplamente utilizada em várias áreas, como visão computacional, processamento de linguagem natural e reconhecimento de fala.

6. **Aprendizado Profundo (Deep Learning)**: É uma técnica de aprendizado de máquina que, segundo (LeCun et al., 2015), utiliza redes neurais com múltiplas camadas para aprender representações complexas de dados. Essa abordagem tem proporcionado avanços significativos em diversas áreas como reconhecimento de padrões, processamento de linguagem natural e identificação de objetos.

Com base nos tipos de aprendizado de máquina discutidos anteriormente, conseguimos sintetizá-los na Figura 2.1.

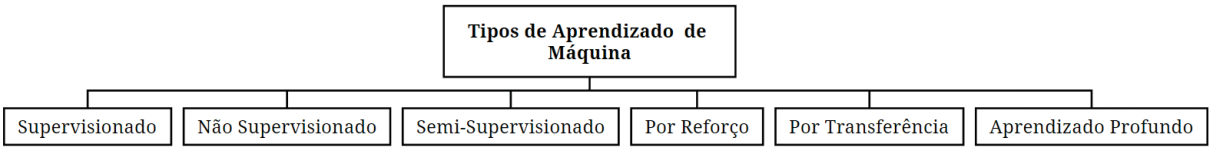


Figura 2.1: Tipos de aprendizado de máquina

A categorização do aprendizado de máquina em supervisionado, não supervisionado, por reforço, semi-supervisionado, por transferência e profundo fornece um arcabouço teórico essencial para a escolha dos algoritmos que serão aplicados no presente trabalho. Cada categoria possui características específicas que as tornam mais adequadas para determinados tipos de problemas. No contexto deste trabalho, o aprendizado supervisionado e o aprendizado profundo são os mais promissores, dada a necessidade de prever eventos com base em dados rotulados.

2.5 ALGORITMOS DE APRENDIZADO DE MÁQUINA

Após a contextualização das categorias de aprendizado de máquina como um alicerce conceitual, direcionamos nossa atenção para as implementações específicas dessas categorias, ou seja, os algoritmos de aprendizado de máquina. A seguir, abordaremos alguns algoritmos, a fim de se definir o algoritmo com maior encaixe para o escopo do trabalho.

- **Regressão Linear:** é uma técnica para modelar a relação entre uma variável dependente e uma ou mais variáveis independentes, de acordo com (Yan e Su, 2009), buscando encontrar a melhor linha reta que descreve essa relação, permitindo prever valores futuros da variável dependente. A regressão linear pode ser simples (com uma variável independente) ou múltipla (com duas ou mais variáveis independentes).

- **Regressão Logística:** é uma técnica estatística usada para prever a probabilidade de um evento binário ocorrer com base em variáveis independentes, conforme (Sainani, 2014). Ela usa uma função logística para modelar a relação entre as variáveis independentes e a variável dependente, permitindo lidar com variáveis categóricas e modelar relações não lineares, sendo utilizada em diversas áreas, incluindo medicina, ciências sociais e negócios, devido à sua capacidade de prever resultados binários.

- **Árvores de Decisão:** é um método amplamente utilizado de aprendizado supervisionado, sendo empregadas tanto para classificação quanto para regressão, já que são capazes de lidar com dados categóricos, numéricos, dados faltantes e são relativamente fáceis de interpretar e visualizar. Elas se estruturam como árvores, onde os nós internos simbolizam atributos ou características, as arestas representam regras de decisão, e cada nó folha é o resultado da decisão, de acordo com (Kotsiantis et al., 2007), no entanto, elas têm algumas limitações, como a tendência de overfitting em conjuntos de dados complexos e a dificuldade em lidar com relações não lineares entre variáveis.

- **Florestas Aleatórias:** é um algoritmo que consiste em um conjunto de árvores de decisão, onde cada árvore é construída a partir de uma amostra aleatória do conjunto de dados de treinamento e uma seleção aleatória de recursos. As florestas aleatórias oferecem alta precisão, robustez, velocidade, estimativas internas úteis e simplicidade de implementação, sendo amplamente utilizadas em problemas de classificação, regressão e detecção de anomalias, de acordo com (Breiman, 2001) .

- **Árvores de Regressão:** são uma técnica de aprendizado de máquina usada para prever valores numéricos, em oposição a valores discretos como nas árvores de classificação. Elas dividem o conjunto de dados em subconjuntos menores com base em variáveis preditoras e preveem valores contínuos, sendo úteis em previsão de preços, vendas e receitas, sendo fáceis de interpretar e ajustar para novos dados, segundo (Loh, 2011) .

- **SVM (Máquinas de Vetores de Suporte):** conforme observado por (Kotsiantis et al., 2007) além da classificação, as SVMs podem ser utilizadas para tarefas de regressão. Segundo o autor elas são capazes de encontrar o hiperplano de separação que maximiza a margem entre as classes, o que ajuda a reduzir o erro de generalização esperado, sendo conhecidas por sua capacidade de lidar com conjuntos de dados de alta dimensionalidade e por sua eficácia em lidar com problemas de classificação binária, tendo algumas limitações, como a baixa velocidade de treinamento e a necessidade de escolher adequadamente o kernel e seus parâmetros.

- **K-Vizinhos Mais Próximos (K-NN):** Segundo (Altman, 1992), é um método de aprendizado de máquina não paramétrico que pode ser utilizado para classificação e regressão. Ele funciona encontrando os K pontos de dados mais próximos de um ponto de consulta e usando a classe ou valor médio desses pontos realiza então previsões. A escolha do valor de K e o custo computacional são considerações importantes ao se utilizar o KNN.

- **Modelos Ocultos de Markov (HMM):** é uma técnica usada em problemas sequenciais, onde estados e transições têm distribuições de probabilidade associadas a emissões de símbolos, sendo aplicado em bioinformática, reconhecimento de fala e processamento de linguagem natural, conforme (Eddy, 1996).

- **K-Means:** é um método popular e eficiente de agrupamento, amplamente utilizado em aplicações como mineração de dados e análise de padrões. De acordo com (Jain et al., 1999), ele funciona interativamente atribuindo pontos de dados a clusters, procurando minimizar a soma dos quadrados das distâncias entre os pontos e os centróides dos clusters. O k-means tem uma complexidade de tempo linear em relação ao número de padrões no conjunto de dados, tornando-o eficiente para conjuntos de dados de grande escala, porém seu desempenho varia em função do lugar de inicialização dos centróides e do número de clusters escolhido.

- **Naïve Bayes:** é frequentemente utilizado para tarefas de classificação, especialmente em problemas de processamento de linguagem natural e mineração de texto, sendo conhecido por sua simplicidade e eficiência computacional, que o tornam adequado para grandes conjuntos de dados, podendo lidar com dados categóricos, numéricos e dados ausentes, conforme (Kotsiantis et al., 2007).

- **Redes Neurais Convolucionais (CNN):** são redes neurais profundas projetadas para processar dados com estrutura de grade, como imagens e sinais de áudio. Elas consistem

em camadas de convolução para extrair características, seguidas por camadas de pooling para reduzir a dimensionalidade. As CNNs aprendem representações hierárquicas de características, tornando-as eficazes em tarefas de reconhecimento de imagem e processamento de dados estruturados, segundo (LeCun et al., 1995).

- **Redes Neurais Recorrentes (RNN):** de acordo com (Medsker e Jain, 2001a) são um tipo de rede neural projetada para lidar com padrões sequenciais ou variáveis no tempo, caracterizadas por conexões de feedback, sendo aplicadas a uma variedade de problemas, desde o aprendizado de sequências de caracteres até a detecção de movimento e previsão financeira. A presença de feedback em RNNs permite a geração de representações internas e dispositivos de memória, tornando-as essenciais para o processamento de informações espaço-temporais.

A escolha do algoritmo de aprendizado de máquina adequado é crucial para o sucesso do modelo proposto. Algoritmos como Florestas Aleatórias e Redes Neurais Recorrentes destacam-se pela sua capacidade de lidar com dados complexos e sequenciais, respectivamente. No entanto, a decisão final deve considerar o equilíbrio entre precisão, eficiência computacional e facilidade de implementação.

2.6 APRENDIZADO PROFUNDO E REDES NEURAIS RECORRENTES

A seleção do uso de técnicas de aprendizado profundo para este projeto se fundamenta na capacidade desses métodos em lidar com a complexidade dos dados provenientes dos sensores de dispositivos móveis. Em particular, o emprego do Deep Learning, que conforme descrito por (LeCun et al., 2015), refere-se a uma abordagem de aprendizado de máquina que se utiliza de redes neurais artificiais com múltiplas camadas para aprender representações de dados em diversos níveis de abstração e que vem apresentando avanços significativos no estado-da-arte em áreas como reconhecimento de fala, detecção de objetos e diversos outros domínios.

No âmbito das redes neurais recorrentes (RNNs), enquanto técnica de aprendizado profundo, conforme descrito por (Medsker e Jain, 2001a) , essas se destacam por sua capacidade em lidar com padrões sequenciais ou variáveis no tempo, caracterizadas por conexões de feedback, sendo aplicadas a uma variedade de problemas, desde o aprendizado de sequências de caracteres até a detecção de movimento e previsão financeira. A presença de feedback em RNNs permite a geração de representações internas e dispositivos de memória, tornando-as essenciais para o processamento de informações espaço-temporais, sendo especialmente adequadas para o processamento de informações em formato de imagem, onde a detecção de padrões é essencial. Estas redes são particularmente relevantes para tarefas como reconhecimento de objetos, segmentação de imagens e classificação de imagens.

Portanto, a escolha de utilizar técnicas de Deep Learning e redes neurais recorrentes se fundamenta na habilidade dessas abordagens em lidar com a natureza complexa e sequencial dos dados provenientes dos sensores, visando à precisão e eficácia na interpretação dessas informações.

2.7 APLICATIVOS EM DISPOSITIVOS MÓVEIS

De acordo com o artigo de (Lane et al., 2010), os dispositivos móveis têm um grande potencial para revolucionar diversas áreas, como negócios, saúde, redes sociais, monitoramento ambiental e transporte, por meio de seu uso como dispositivos de sensoriamento. Levando isso em consideração, escolhemos dispositivos móveis como a solução para o presente trabalho, não

apenas por serem acessíveis à maioria das pessoas, como também por possuírem um grande potencial inexplorado.

A utilização de dispositivos móveis como plataforma para coleta de dados e execução de modelos de aprendizado de máquina oferece uma solução acessível e escalável. No entanto, é necessário considerar as limitações de hardware e a variabilidade na disponibilidade de sensores entre diferentes dispositivos.

2.8 ANÁLISE EXPLORATÓRIA DE DADOS

A análise e tratamento dos dados é uma etapa fundamental, visto que são os dados que irão alimentar o modelo de aprendizado de máquina que será criado e, sendo assim, a qualidade de predição depende significativamente da qualidade dos dados que serão utilizados. No presente trabalho, utilizaremos três bibliotecas do Python para a análise, limpeza de dados, tratamento de valores ausentes e a normalização: Pandas, NumPy e Matplotlib.

Para (Nelli, 2015), Pandas, NumPy e Matplotlib são bibliotecas fundamentais para a análise de dados em Python. Pandas é uma biblioteca que fornece estruturas de dados de alto desempenho e de fácil utilização, além de ferramentas de análise de dados, construída sobre a biblioteca NumPy. NumPy, por sua vez, fornece suporte para arrays e matrizes multidimensionais, bem como funções matemáticas para trabalhar com esses arrays. Já Matplotlib é uma biblioteca de visualização de dados de código aberto que fornece uma ampla variedade de gráficos e visualizações.

A análise exploratória de dados é uma etapa crucial para garantir a qualidade dos dados que alimentarão o modelo. O uso de bibliotecas como Pandas, NumPy e Matplotlib permite uma análise detalhada e a identificação de padrões relevantes, que serão essenciais para o desenvolvimento do modelo classificatório.

2.9 DESENVOLVIMENTO DO MODELO DE APRENDIZADO DE MÁQUINA

Para o desenvolvimento do modelo de aprendizado de máquina, assim como para o desenvolvimento de aplicativos para dispositivos móveis, existem inúmeras possibilidades entre linguagens de programação e bibliotecas, mas novamente nos limitaremos a utilização da linguagem de programação Python e uma das bibliotecas Scikit-learn, TensorFlow, PyTorch e Keras, descritas a seguir.

Segundo (Fazeen et al., 2012) o Scikit-learn apresenta diversas vantagens significativas, tais como implementações robustas de algoritmos de aprendizado de máquina, foco na qualidade do código por meio de testes unitários e análise estática, facilidade de uso, desempenho e consistência da API, distribuição sob a licença BSD, integração com o ecossistema científico do Python, disponibilidade em várias plataformas, incluindo Windows e POSIX, eficiência computacional e implementação de algoritmos de aprendizado supervisionado e não supervisionado.

Já o TensorFlow, como uma biblioteca de aprendizado de máquina desenvolvida pela Google, de acordo com (Abadi et al., 2016), destaca-se como uma ferramenta robusta e versátil para o desenvolvimento em larga escala. Sua flexibilidade e escalabilidade, possibilitadas pelo uso de grafos de fluxo de dados, oferecem aos desenvolvedores a capacidade de experimentar com otimização e algoritmos de treinamento. Além disso, o suporte a uma variedade de dispositivos, como CPUs multicore, GPUs e TPUs, demonstra a adaptabilidade do TensorFlow a diferentes ambientes computacionais. A capacidade de realizar treinamento e inferência em grande escala, utilizando servidores, juntamente com o respaldo de uma comunidade ativa desde seu lançamento

como software de código aberto, realça o potencial do TensorFlow como uma ferramenta integral para pesquisa, experimentação e produção em aprendizado de máquina em larga escala.

O Pytorch, conforme (Paszke et al., 2019), se destaca por sua facilidade de uso e desempenho, por ter sido construído em torno da linguagem Python, também se integra naturalmente com outras bibliotecas populares de computação científica do Python, sendo projetado para tornar a escrita de modelos, carregadores de dados e otimizadores o mais fácil e produtivo possível, escondendo a complexidade inerente ao aprendizado de máquina por trás de uma API intuitiva.

O Keras é uma API versátil que suporta múltiplos frameworks, como JAX, TensorFlow e PyTorch. Segundo (Keras, 2024), essa abordagem oferece vantagens significativas, permitindo a escolha dinâmica do backend para otimizar o desempenho do modelo, também facilitando a máxima compatibilidade com ecossistemas, possibilitando que modelos sejam usados em diferentes frameworks, ampliando a distribuição de modelos de código aberto. A flexibilidade do Keras também se estende à manipulação de dados, sendo compatível com diversas fontes, como tf.data.Dataset, PyTorch DataLoader, NumPy arrays e Pandas dataframes, o que simplifica o desenvolvimento e a implementação de modelos de aprendizado profundo e promove uma abordagem unificada e eficiente.

Um estudo comparativo entre as principais bibliotecas para aprendizado de máquina, realizado por (Gevorkyan et al., 2019), mediu o tempo de treinamento e a precisão do classificador para o reconhecimento de dígitos manuscritos, da base de dados MNIST, criado em cada biblioteca. O TensorFlow apresentou uma precisão de 89,07% com um tempo de treinamento rápido em comparação com as demais bibliotecas (63,48 segundos), enquanto o PyTorch apresentou uma precisão superior de 98,07%, mas com um tempo de treinamento significativamente maior (1492,29 segundos), justificado pelo suporte a um gráfico computacional dinâmico. Já o SciKit Learn demonstrou uma precisão de 93,45% com um tempo de treinamento intermediário (146,9 segundos) e Keras apresentou uma precisão de 98,56% e um tempo de treinamento intermediário (113,80 segundos).

Por fim, este trabalho utilizou o TensorFlow para o treinamento dos modelos, destacando-se por seu tempo de treinamento reduzido em comparação com outras bibliotecas, o que permitiu o treinamento de múltiplos modelos diferentes, com diferentes váriaveis e condições de treinamento. Além disso, a integração com o TensorFlow Lite, que é uma extensão do TensorFlow, facilitou a conversão do modelo treinado para um formato otimizado para dispositivos móveis, garantindo um processo mais simples, rápido e eficiente.

Adicionalmente, ferramentas da biblioteca Scikit-learn foram empregadas para a normalização dos dados, bem como para sua divisão em conjuntos de treinamento (train), teste (test) e validação (validation), etapas fundamentais para assegurar a robustez e a confiabilidade dos modelos desenvolvidos.

A escolha da biblioteca adequada para o desenvolvimento do modelo de aprendizado de máquina é crucial para o sucesso do projeto. O TensorFlow e o PyTorch destacam-se pela sua flexibilidade e eficiência, enquanto o Scikit-learn é ideal para modelos mais simples e rápidos de implementar.

2.10 TRABALHOS RELACIONADOS

A partir da exploração de outros trabalhos relacionados foi possível prever eventuais empecilhos e soluções para o presente trabalho. Um exemplo disso foi o trabalho de (Nasreen et al., 2019), no qual se demonstra o impacto de diferentes níveis de iluminação na acurácia

do modelo de aprendizado de máquina, além de também apontar a interferência do modelo de celular utilizado na captura da imagem para o nível de acurácia obtido.

Diversos estudos exploraram a integração de smartphones e algoritmos de aprendizado de máquina para aprimorar aspectos do transporte e da infraestrutura viária. (Alvarez et al., 2014) propuseram um sistema de estimativa de carga restante em veículos elétricos, utilizando redes neurais. As redes foram treinadas em uma rota específica em Madri, Espanha, utilizando dados de velocidade e aceleração coletados por meio de smartphones, bem como valores de consumo extraídos diretamente de uma instrumentação a bordo do veículo elétrico de 10 motoristas diferentes. O sistema de estimativa resultante do treinamento, voltado para Sistemas de Transporte Inteligentes (ITS), destaca a eficácia dos sensores de smartphones na estimativa de carga, proporcionando informações com mais de 95% de precisão aos motoristas.

Outro estudo relevante, conduzido por (Tiwari et al., 2020), apresentou o RoadCare, uma solução de crowdsourcing para monitorar a qualidade da superfície das estradas em áreas urbanas. Para criar esta solução, um modelo baseado em deep learning foi treinado utilizando dados coletados por meio de uma aplicação Android desenvolvida, ao longo de 36 horas, em diferentes tipos de estradas. O estudo destaca a precisão superior a 98% alcançada pelo RoadCare para a classificação, de imagens de seções de estradas em uma escala de 11 pontos, e destaca seu potencial para fornecer alertas antecipados aos usuários de estradas, demonstrando a aplicabilidade prática de dados de smartphones e algoritmos de aprendizado profundo na avaliação da qualidade das estradas.

(Basavaraju et al., 2019) exploraram a utilização de dados provenientes de acelerômetro, giroscópio e GPS de smartphones para a classificação precisa das condições da estrada. Com o objetivo de melhorar a experiência de direção e minimizar acidentes, o estudo empregou técnicas de aprendizado de máquina, incluindo redes neurais profundas que foram treinadas utilizando uma base de dados gerada por meio de smartphones. A base de dados consiste em 1010 amostras, cada uma contendo um vetor de características e um rótulo de condição da estrada. Os resultados indicaram que a utilização de dados dos três eixos dos sensores do smartphone resulta em maior precisão na classificação, resultando em uma acurácia de mais de 90% e assim evidenciando o potencial dessas abordagens para monitorar estradas, identificar defeitos e oferecer informações de manutenção.

O artigo de (Fazeen et al., 2012) propõe o uso de smartphones equipados com acelerômetros de três eixos para registrar e analisar comportamentos do motorista e condições da estrada, visando aumentar a conscientização e maximizar a segurança no trânsito. Os objetivos incluem utilizar feedback auditivo em tempo real, oferecer uma solução de baixo custo e portátil, demonstrar a eficácia dos smartphones como dispositivos de medição e identificar comportamentos perigosos para aprimorar sistemas de assistência ao motorista. Resultados incluem a identificação precisa de comportamentos de direção e anomalias na estrada, juntamente com feedback auditivo eficaz. Os resultados quantitativos indicam alta precisão na classificação de buracos (85,6%), solavancos (81,5%) e estradas lisas (100%), mostrando assim que os smartphones são eficazes para melhorar a segurança no trânsito e a conscientização do motorista.

Os trabalhos demonstram a viabilidade e o potencial da integração de sensores móveis e algoritmos de aprendizado de máquina para a melhoria da infraestrutura viária e da segurança no transporte. No entanto, é necessário considerar as limitações e desafios identificados nesses estudos, como a variabilidade na qualidade dos dados coletados e a necessidade de modelos robustos e generalizáveis.

3 METODOLOGIA

O objetivo geral deste trabalho consistiu em desenvolver um aplicativo para dispositivos móveis Android, voltado para a comunidade de ciclistas, visando não apenas melhorar a qualidade de vida e segurança dos ciclistas, como também contribuir para o desenvolvimento sustentável da infraestrutura viária urbana. Para alcançar essa meta, o projeto propôs a integração de tecnologias, como os sensores dos dispositivos móveis e algoritmos de aprendizado de máquina, para capturar dados em tempo real sobre as condições das vias.

Esta coleta de dados foi viabilizada por meio do desenvolvimento de um aplicativo em Flutter que coletou dados para o treinamento e abastecimento do modelo de aprendizado de máquina. A execução desse objetivo geral se deu por partes, orientada por objetivos específicos que serão detalhados a seguir.

3.1 OBJETIVO GERAL

Desenvolver um modelo de aprendizado de máquina classificatório, utilizando dados coletados por sensores de dispositivos móveis, para prever a qualidade das vias e fornecer informações úteis tanto para ciclistas quanto para entidades públicas responsáveis pela infraestrutura urbana.

3.2 OBJETIVOS ESPECÍFICOS

- **Desenvolvimento do aplicativo de coleta de dados:** Visando a coleta eficiente de dados provenientes de sensores presentes em dispositivos móveis, como acelerômetro, giroscópio, GPS, câmera e quaisquer outros sensores que apresentem relevância para o presente trabalho, desenvolvemos um aplicativo para as plataformas Android e iOS. Para a construção do aplicativo, optou-se pela linguagem de programação Flutter, devido à sua capacidade de desenvolvimento multiplataforma e à facilidade de integração com bibliotecas de sensores. O aplicativo foi projetado para coletar dados em tempo real e armazená-los em arquivos CSV no próprio dispositivo, facilitando a análise posterior. A coleta de dados foi realizada em diferentes condições, variando a posição do dispositivo e o tipo de bicicleta, para garantir a generalização do modelo.

- **Análise exploratória de dados:** Utilizamos Jupyter Notebooks e bibliotecas como Pandas, NumPy e Matplotlib para conduzir a análise exploratória dos dados obtidos por meio do aplicativo desenvolvido. Durante essa análise, realizamos o pré-processamento e a rotulação dos dados, garantindo que estivessem no formato apropriado para alimentar o modelo em uma fase subsequente. Essa etapa envolveu:

  - **Limpeza de dados:** Identificação e tratamento de valores ausentes, utilizando técnicas como imputação pela média ou exclusão de registros incompletos.
  - **Normalização:** Aplicação da escala Min-Max para garantir que todas as variáveis estivessem no mesmo intervalo [0, 1], facilitando a comparação e o processamento.
  - **Rotulação manual:** Os dados foram rotulados manualmente com base em vídeos gerados durante a coleta, que exibiam o trajeto percorrido e o horário correspondente

a cada segundo. Um script em Python foi desenvolvido para automatizar a inserção de timestamps nos vídeos, facilitando a rotulação.

  - **Identificação de padrões:** Foram calculadas médias móveis para as variáveis dos sensores, utilizando janelas de 50, 100 e 200 amostras, para capturar padrões de variação ao longo do tempo. Razões entre as médias móveis e a velocidade, bem como diferenças entre médias consecutivas, foram incorporadas para enriquecer o conjunto de dados.

- **Desenvolvimento do modelo de aprendizado de máquina:** Partindo da revisão bibliográfica, múltiplos algoritmos de aprendizado de máquina, como Redes Neurais, Florestas Aleatórias, Árvores de Decisão e Regressão Logística, foram escolhidos e aplicados. O processo de desenvolvimento do modelo incluiu as seguintes etapas:

  - **Divisão dos dados:** Os dados foram divididos em três conjuntos: treino (60%), teste (20%) e validação (20%). Essa divisão foi realizada utilizando a função `train_test_split` da biblioteca Scikit-learn, garantindo uma distribuição equilibrada das classes em cada conjunto.
  - **Seleção de variáveis:** Após testes iniciais, foram selecionadas as variáveis mais relevantes.
  - **Ajuste de hiperparâmetros:** Foram testadas diferentes configurações de camadas e taxas de aprendizado para o modelo de Redes Neurais, utilizando a biblioteca TensorFlow.
  - **Avaliação dos modelos:** Os modelos foram avaliados utilizando métricas como acurácia, precisão, recall e F1-score.

3.3 DESENVOLVIMENTO DO APLICATIVO

Para o desenvolvimento do aplicativo em questão, utilizamos o Software Android Studio em conjunto com a linguagem de programação Flutter, baseada em Dart. Essa escolha foi justificada pela compatibilidade do Flutter com múltiplas plataformas, como Android e iOS, assim abrangendo a maior parcela dos usuário.

Além disso, aproveitamos as seguintes bibliotecas para aprimorar a funcionalidade do aplicativo:

- **cupertino_icons** (versão 1.0.2): Fornece ícones no estilo Cupertino (usado no iOS).
- **sensors_plus** (versão 4.0.2): Facilita o acesso aos sensores do dispositivo.
- **location** (versão 5.0.3): Permite obter informações de localização.
- **camerawesome** (versão 2.0.1): Uma biblioteca para trabalhar com a câmera do dispositivo.
- **camera** (versão 0.10.5+9): Outra biblioteca para funcionalidades relacionadas à câmera.
- **open_file** (versão 3.3.2): Ajuda a abrir arquivos no sistema.
- **flutter_background_service** (versão 5.0.5): Permite executar tarefas em segundo plano.
- **permission_handler** (versão 11.3.0): Gerencia permissões do aplicativo.

- **image_gallery_saver** (versão 2.0.3): Salva imagens na galeria.
- **gallery_saver** (versão 2.3.2): Outra opção para salvar imagens.
- **csv** (versão 6.0.0): Manipula arquivos CSV.
- **file_picker** (versão 6.1.1): Permite selecionar arquivos do dispositivo.
- **awesome_dialog** (versão 3.2.0): Cria diálogos personalizados.
- **shared_preferences** (versão 2.2.2): Armazena dados persistentes no aplicativo.
- **fl_chart** (versão 0.68.0): Biblioteca para gráficos.
- **get** (versão 4.6.6): Gerenciamento de estado simplificado.
- **share_plus** (versão 9.0.0): Compartilha conteúdo com outras aplicações.
- **circular_countdown_timer** (versão 0.2.3): Exibe um contador circular.
- **geolocator** (versão 12.0.0): Fornece informações de geolocalização.

3.4 CONSIDERAÇÕES FINAIS SOBRE A METODOLOGIA

A metodologia adotada neste trabalho permitiu o desenvolvimento de um modelo classificatório robusto e eficiente, capaz de prever a qualidade das vias com alta precisão. Futuros trabalhos podem explorar a ampliação do conjunto de dados e a integração com outras plataformas, como sistemas de mapeamento, para aprimorar ainda mais a aplicação.

4 RESULTADOS

4.1 COLETA DOS DADOS

Para a coleta dos dados utilizados no treinamento dos modelos, o celular foi posicionado em diversas configurações em diferentes modelos de bicicletas. Essa abordagem teve como objetivo ampliar a generalização do modelo, condicionando-o a reconhecer uma ampla variedade de cenários e situações, incluindo as variações inerentes a diferentes tipos de bicicletas e posições do dispositivo. Isso garantiu que o modelo fosse robusto e capaz de lidar com a diversidade presente no uso real. O montante total de dados coletados foi de 453.951 amostras.

4.2 PREPROCESSAMENTO DOS DADOS

Os dados foram coletados por meio do aplicativo desenvolvido, resultando em uma forma de dados inicialmente adequada para análise. Portanto, não foi necessário realizar quaisquer transformações ou procedimentos adicionais de limpeza.

4.3 ROTULAÇÃO DOS DADOS

Para facilitar a rotulação manual dos dados coletados, este trabalho implementou um processo que gera automaticamente um vídeo de todo o trajeto percorrido em cada execução do código. Para aprimorar a experiência de rotulação, desenvolveu-se um script que insere uma legenda no vídeo, exibindo o horário correspondente a cada segundo do trajeto. Com isso, foi possível realizar a rotulação dos dados de maneira mais eficiente, bastando informar o horário inicial, o horário final e o rótulo correspondente enquanto se assistia ao vídeo.

O script utiliza a biblioteca MoviePy para adicionar os timestamps como legendas no vídeo, extraindo o horário base do nome do arquivo e atualizando-o dinamicamente para cada frame, conforme a duração do vídeo. Dessa forma, o vídeo resultante exibe os horários de forma sincronizada com o trajeto percorrido, facilitando a identificação precisa dos intervalos de interesse.

Complementarmente, um segundo script foi implementado para processar os dados do sensor. Ele gera uma coluna de tempos calculados no formato HH:MM:SS.mmm, baseada no intervalo de tempo registrado entre as amostras. Este script também permite rotular os dados diretamente em um arquivo, oferecendo uma interface interativa para definir horários de início, fim e os respectivos rótulos, que são então armazenados no arquivo final para análises posteriores.

4.4 ANÁLISE E NORMALIZAÇÃO

Devido à alta frequência de amostragem dos dados, com registros a cada 20 milissegundos, foi adotada a estratégia de cálculo de médias móveis utilizando múltiplas janelas temporais. Essa abordagem permite integrar informações históricas com os dados mais recentes, promovendo uma análise mais robusta e sensível às variações dinâmicas do sistema. Para cada variável, foram calculadas médias móveis com janelas de 50, 100 e 200 amostras. Além disso, razões entre as médias móveis e a velocidade, bem como as diferenças entre médias consecutivas, foram incorporadas para capturar padrões de variação e correlações importantes.

Figura 4.1: Tela inicial do aplicativo com três botões principais: Executar, Mais Informações e Configurações. Executar, leva o usuário a tela de coleta de dados, Informações, conduz o usuário a uma tela com mais dados do aplicativo, Configurações, permite o usuário habilitar e desabilitar sensores.

Figura 4.2: Tela das configurações dos sistema proposto, é essencial para garantir flexibilidade na coleta de dados, permitindo que o usuário habilite ou desabilite sensores específicos conforme necessário.

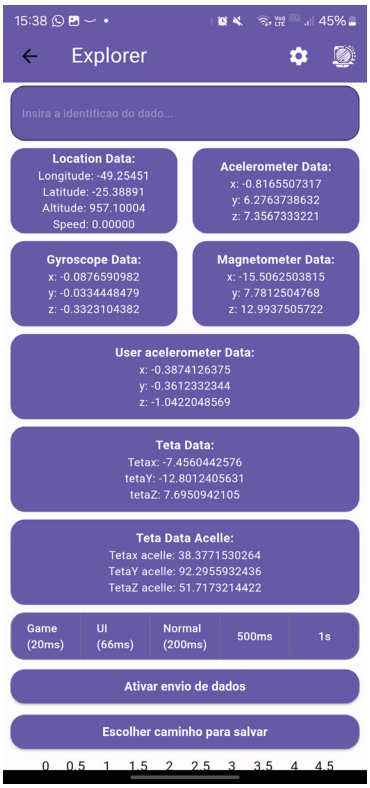


Figura 4.3: Tela de captura de dados - representa a interface do aplicativo desenvolvido para a coleta de informações durante os percursos realizados pelos ciclistas. Esta tela permite que o usuário visualize em tempo real as informações provenientes de cada sensor do dispositivo móvel utilizado. Os principais elementos observados incluem: Visualização em tempo real dos dados dos sensores: O usuário pode acompanhar as leituras instantâneas de todos os sensores ativados durante o percurso, possibilitando um monitoramento direto das condições da via. Configuração do intervalo de captura: Há a possibilidade de definir a frequência com que os dados são registrados, permitindo ajustar a granularidade da coleta conforme necessário. Definição da rotulação: A interface oferece a opção de atribuir rótulos aos dados coletados, facilitando a posterior categorização e análise das informações. Escolha da pasta de armazenamento: O usuário pode determinar onde os dados serão salvos no dispositivo, proporcionando organização e controle sobre o gerenciamento dos arquivos gerados. Esta figura ilustra a funcionalidade principal do aplicativo desenvolvido para a pesquisa, demonstrando como os dados brutos das condições das vias urbanas eram coletados diretamente durante os trajetos de bicicleta. Os dados capturados por esta interface serviram como base para todo o processamento e análise posterior, culminando no desenvolvimento do modelo

No contexto da heterogeneidade das escalas das variáveis — decorrente de diferentes unidades de medida —, foi essencial aplicar um processo de normalização utilizando a escala Min-Max. Inicialmente, ausentes foram substituídos por valores finitos ou imputados com base na média. Em seguida, todas as variáveis numéricas foram ajustadas para um intervalo comum [0, 1], preservando a variância relativa entre os dados e facilitando comparações diretas, análises estatísticas e modelagem classificatória.

4.5 COMPARAÇÃO DOS MODELOS CLASSIFICATÓRIO E CLASSIFICAÇÃO COM CONJUNTO DE DADOS DE TESTE

Com os dados devidamente mapeados na etapa anterior o modelo classificatório pode ser criado. Os modelos escolgidos para teste foram:

- Arvore de decisão;
- Gaussian Naive Bayes;
- Gradient Bossting Classifier;
- Floresta aleatória;
- Regressão Logistica;
- Redes Neurais.

Para definir qual modelo seria utilizado separamos a base de dados em 3 bases, 1 para o treinamento do modelo, 1 para ajustar seus hiperparametros e uma terceira para validar o modelo e obter estatisticas de precisão, acurácia, recall e F1. Os resultados obtidos são demonstrados na tabela 4.1 e figura 4.4.

As métricas a ser priorizada foi o F1-score, visto que é uma métrica balanceada que combina precisão e recall, sendo ideal para situações onde o equilíbrio entre identificar corretamente tanto os positivos quanto os negativos é importante. Oferecendo uma visão abrangente da capacidade do modelo em lidar com diferentes tipos de estradas de forma precisa e eficiente.

Modelo	Accuracy	Precision	Recall	F1 Score
Redes Neurais	0.9903	0.9904	0.9903	0.9903
Árvore de Decisão	0.9994	0.9994	0.9994	0.9994
Gaussian Naive Bayes	0.8252	0.8206	0.8252	0.8181
Gradient Boosting Classifier	0.9788	0.9788	0.9788	0.9785
Floresta Aleatória	0.9999	0.9999	0.9999	0.9999
Regressão Logística	0.8263	0.8100	0.8263	0.8137

Tabela 4.1: Métricas de desempenho dos modelos - A Random Forest obteve o melhor resultado geral, com todas as métricas próximas de 1. Árvores de Decisão e Redes Neurais também mostraram ótimo desempenho. Já o Gaussian Naive Bayes e a Regressão Logística tiveram resultados inferiores em todas as métricas. O Gradient Boosting Classifier apresentou um equilíbrio, com desempenho próximo das melhores técnicas. Os modelos mais complexos tiveram maior precisão e recall.

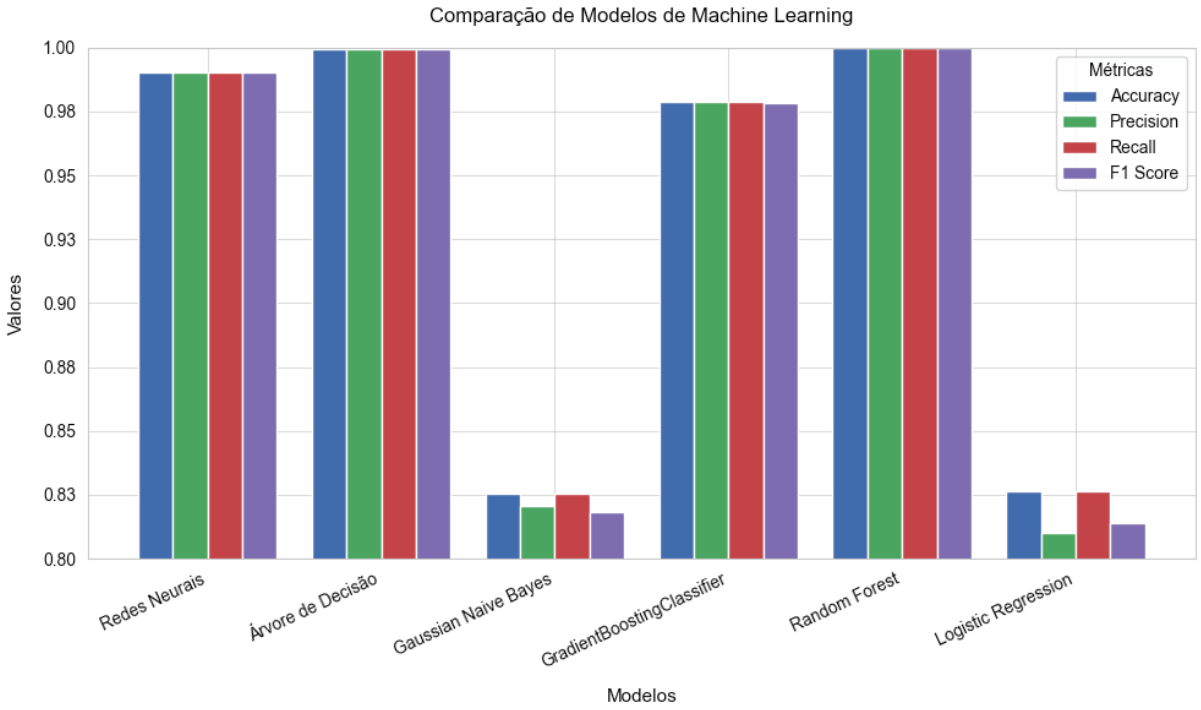


Figura 4.4: Comparação de modelos de Aprendizado de Máquina - Os modelos de Redes Neurais, árvore de decisão, Gradient Boosting Classifier e Random Forest apresentam resultados significativos, com precisão, acurácia, recall e F1 score superior a 96%, entre os 4 modelos o modelo de Random Forest é o que mais se destaca nos 4 indicadores.

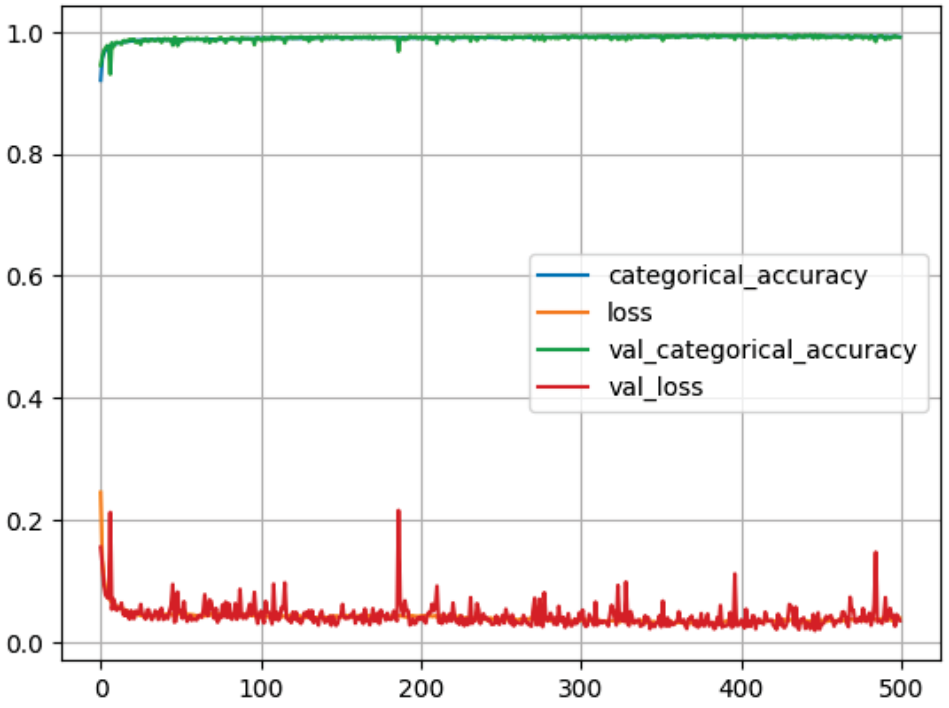


Figura 4.5: Desempenho do modelo de Redes Neurais - mostra o desempenho do modelo em 500 épocas. As linhas azul e verde representam a precisão para treino e validação, ambas próximas de 1, indicando alta acurácia. A perda de treino (laranja) é baixa e estável, enquanto a perda de validação (vermelho) apresenta oscilações. A estabilidade da precisão indica bom aprendizado geral.

4.6 AVALIAÇÃO DOS MODELOS CLASSIFICATÓRIOS

Para avaliar a eficácia dos modelos de *Aprendizado de Máquina* neste estudo, adotou-se uma abordagem estruturada de divisão dos dados. Inicialmente, os dados foram separados em conjuntos de treinamento completo (`X_train_full`, `y_train_full`) e predição (`X_prediction`, `y_prediction`) usando a função `train_test_split` do `scikit-learn`, com 80% dos dados destinados ao treinamento completo e 20% reservados para avaliação final do modelo. Após essa divisão, o conjunto `X_prediction` continha 90.791 amostras.

Em seguida, o conjunto de treinamento completo foi subdividido, novamente utilizando a função `train_test_split`, em conjuntos de treinamento (`X_train`, `y_train`) e validação (`X_test`, `y_test`), correspondendo a 75% e 25%, respectivamente, do conjunto de treinamento completo. Isso resultou em 272.370 amostras no conjunto `X_train` e 90.790 amostras no conjunto `X_test`.

Adicionalmente, os rótulos foram convertidos para o formato *one-hot encoding* utilizando a função `to_categorical`, garantindo compatibilidade com os modelos de *Aprendizado de Máquina* utilizados.

Após a divisão dos dados, procedeu-se com o treinamento de seis modelos de *Aprendizado de Máquina* diferentes: Redes Neurais, Árvore de Decisão, Gaussian Naive Bayes, Gradient Boosting Classifier, Floresta Aleatória e Regressão Logística. Cada modelo foi treinado usando o conjunto `X_train` e avaliado usando métricas padrão como *acurácia*, *precisão*, *recall* e F1-*score*. Essas métricas foram calculadas tanto no conjunto de validação (`X_test`) quanto no conjunto de predição (`X_prediction`), permitindo uma comparação robusta do desempenho dos modelos em diferentes conjuntos de dados.

4.7 TREINAMENTO DO MODELO DE REDES NEURAIS

O treinamento do modelo de redes neurais foi conduzido em etapas iterativas, explorando diferentes configurações de entrada, arquiteturas e hiperparâmetros. Inicialmente, foram testados diversos modelos variando a janela de dados utilizada. As médias móveis de 25, 50, 100 e 200 amostras foram consideradas, tanto com quanto sem as variáveis de altitude e velocidade, além de incluir ou não a divisão pela velocidade e a subtração em relação à janela anterior. Após extensivos testes, a melhor configuração inicial encontrada utilizou uma média móvel de 100 amostras, incluindo a velocidade, a ponderação pela velocidade e a diferença em relação à janela anterior. Esses testes resultaram no treinamento de 48 modelos.

Com base nesses resultados, a análise foi refinada, fixando a média móvel de 100 amostras e avaliando variantes específicas: com ou sem ponderação pela velocidade e com ou sem a subtração em relação à janela anterior. Entre os 8 modelos testados nesta etapa, o melhor apresentou um F1 score de 0,989 ao considerar a subtração em relação à janela anterior. No entanto, a diferença marginal de 0,004 no F1 score em comparação com um modelo mais simples (F1 de 0,985), que não incluía nenhuma dessas variáveis adicionais, levou à escolha deste último. Esse modelo utilizava 18 variáveis de entrada:

- `gyroscopeZ_mov_avg_100`
- `gyroscopeY_mov_avg_100`
- `gyroscopeX_mov_avg_100`
- `tetaX_mov_avg_100`

- `tetaY_mov_avg_100`
- `tetaZ_mov_avg_100`
- `acelerometer_raw_z_mov_avg_100`
- `acelerometer_raw_y_mov_avg_100`
- `acelerometer_raw_x_mov_avg_100`
- `tetaX_accelerometer_mov_avg_100`
- `tetaY_accelerometer_mov_avg_100`
- `tetaZ_accelerometer_mov_avg_100`
- `acelerometer_z_mov_avg_100`
- `acelerometer_y_mov_avg_100`
- `acelerometer_x_mov_avg_100`
- `gyro_magnitude_mov_avg_100`
- `accel_magnitude_mov_avg_100`
- `speed`

Para reduzir ainda mais a complexidade do modelo, as combinações dessas variáveis foram avaliadas. O mínimo de 2 e o máximo de 8 variáveis por combinação foram testados, respeitando agrupamentos por categorias (ex.: acelerômetro e giroscópio). Isso resultou em 247 combinações e, consequentemente, 247 modelos treinados, em um processo que levou aproximadamente 10 horas e meia. Após filtrar os modelos para aqueles com até 6 variáveis, o melhor resultado foi obtido com F1 de 0,977, utilizando as seguintes variáveis:

- `accel_magnitude_mov_avg_100`
- `speed`
- `tetaX_mov_avg_100`
- `tetaY_mov_avg_100`
- `tetaZ_mov_avg_100`

Embora tenha ocorrido uma perda de 0,008 no F1 score em relação ao modelo anterior, essa redução foi considerada aceitável devido à simplificação significativa (redução para um terço das variáveis).

Com as variáveis reduzidas, iniciou-se o ajuste da arquitetura do modelo e da taxa de aprendizado (*learning rate*). Foram testadas as seguintes configurações de camadas:

- `[36, 12]` (configuração original)
- `[64, 36, 12]` (adicionando camadas)

- `[36]` (removendo uma camada)
- `[14]` (simplificando)
- `[17, 8]` (configuração reduzida)
- `[128, 64, 36, 12]` (aumentando a profundidade)

E as seguintes taxas de aprendizado: `[0.001, 0.01, 0.1]`. Isso resultou no treinamento de 18 modelos. O melhor modelo apresentou a configuração `[64, 36, 12]` com uma *learning rate* de 0,001, alcançando um F1 score de 0,9845 em 25 *epochs*.

Por fim, o modelo final foi treinado com essas configurações otimizadas por 500 *epochs*, resultando em um F1 score de 0,9902, consolidando-o como a melhor solução obtida no experimento. Fruto do treinamento de um total de 321 modelos diferentes.

4.8 ESCOLHA DE UM MODELO

Com base nas métricas apresentadas na Tabela 4.1 e na figura 4.4, observa-se que o modelo Florestas Aleatórias obteve um desempenho excepcionalmente alto em todas as métricas avaliadas: Accuracy, Precision, Recall e F1 Score, todos registrados em 0,999.

Apesar do modelo de redes neurais não ter sido definido como o modelo final todo seu processo de construção e desenvolvimento foi essencial para a definição das váriaveis finais que foram então utilizadas para o treinamento de todos os demais modelos.

A escolha do modelo de Florestas aleatórias como o modelo destacado é justificada pela seu desempenho uniformemente alto em todas as métricas indica uma consistência robusta em diferentes conjuntos de dados, o que é um indicativo positivo de sua adaptabilidade e eficácia.

Portanto, com base nas métricas de desempenho apresentadas e na sua capacidade de manter um alto F1-score de 0,999, Florestas aleatórias é escolhida como o modelo mais adequado para este estudo. Ela não apenas demonstraram um excelente desempenho geral, mas também são capazes de lidar eficazmente com a variedade de condições de estrada apresentadas nos dados coletados.

5 TRABALHOS FUTUROS

Embora os resultados obtidos neste trabalho demonstrem a viabilidade e eficácia da aplicação de aprendizado de máquina para melhorar a experiência e a segurança dos ciclistas, diversas oportunidades para aprimoramentos e extensões ainda podem ser exploradas. Dentre os possíveis futuros trabalhos, destacam-se:

- **Treinamento de modelos em larga escala:** A ampliação do conjunto de dados com coletas realizadas em diferentes cidades e contextos geográficos pode melhorar a generalização do modelo, tornando-o aplicável a uma maior diversidade de cenários.
- **Integração com outras plataformas:** O aplicativo pode ser expandido para incluir integração com plataformas de mapeamento, como Google Maps, para exibir rotas otimizadas em tempo real com base na qualidade da infraestrutura viária.
- **Uso em veículos autônomos:** A adaptação do modelo para veículos autônomos pode contribuir para a identificação de obstáculos e melhorias no planejamento de rotas, integrando soluções para transporte de ciclistas e automóveis.
- **Implementação de um sistema de feedback em tempo real:** Desenvolver funcionalidades que permitam aos ciclistas receber alertas em tempo real sobre perigos detectados nas vias, como buracos ou irregularidades, aumentando a segurança durante o trajeto.
- **Exploração de técnicas de aprendizado contínuo:** Implementar métodos de aprendizado contínuo (*online learning*) para permitir que o modelo se ajuste dinamicamente com base em novos dados coletados durante o uso do aplicativo.
- **Colaboração com entidades públicas:** Estabelecer parcerias com órgãos governamentais para utilizar os dados coletados como subsídio para ações de manutenção e planejamento urbano, ampliando o impacto social do projeto.
- **Estudo de impacto social e ambiental:** Realizar uma análise detalhada dos impactos sociais e ambientais do uso do aplicativo, com foco em indicadores como redução de acidentes, aumento da segurança e incentivo ao uso da bicicleta como meio de transporte sustentável.

Essas direções futuras não apenas complementam o escopo do trabalho apresentado, mas também reforçam o potencial do aprendizado de máquina e das tecnologias móveis na promoção de soluções inovadoras e de impacto para a mobilidade urbana sustentável.

6 CONCLUSÃO

Este trabalho demonstrou a viabilidade e o potencial do uso de sensores de dispositivos móveis em conjunto com algoritmos de aprendizado de máquina para a análise e identificação da qualidade de infraestrutura viária, com foco no ciclismo. Desde a coleta de dados até o desenvolvimento de modelos classificatórios, cada etapa foi cuidadosamente planejada e executada, culminando em um modelo robusto e eficiente que pode ser integrado a um aplicativo móvel funcional.

A análise exploratória de dados revelou a importância de estratégias como a utilização de médias móveis e normalização Min-Max para tratar dados heterogêneos provenientes de diferentes sensores. Além disso, o processo de seleção de variáveis mostrou-se essencial para a simplificação do modelo, reduzindo a complexidade sem comprometer significativamente o desempenho. Essas etapas garantiram que os dados estivessem bem preparados para alimentar os modelos de aprendizado de máquina, maximizando sua eficácia.

Os testes comparativos realizados com diferentes algoritmos de aprendizado de máquina evidenciaram a eficácia de modelos como Florestas Aleatórias e Redes Neurais na identificação de condições viárias. Apesar de o modelo Random Forest apresentar métricas de desempenho superiores, o desenvolvimento e ajuste do modelo de Redes Neurais foram fundamentais para a definição das variáveis mais relevantes, destacando a contribuição desse processo para os resultados finais.

Para o cenário em questão, os experimentos realizados demonstraram a superioridade do modelo Random Forest, que alcançou métricas excepcionais com Accuracy, Precision, Recall e F1 Score de 0,999 . As Redes Neurais também apresentaram desempenho expressivo, obtendo um F1 Score de 0,990 após extensivo ajuste de hiperparâmetros e seleção de variáveis. Esses resultados quantitativos comprovam a eficácia da abordagem proposta para identificação das condições das vias urbanas em tempo real.

Além disso, o desenvolvimento de um aplicativo móvel em Flutter permitiu não apenas a coleta eficiente de dados, mas também a criação de uma base sólida para futuras aplicações, como a integração de um modelo classificatório para fornecer informações em tempo real aos ciclistas. A abordagem multiplataforma do Flutter, aliada à compatibilidade com bibliotecas específicas, garantiu a flexibilidade e escalabilidade necessárias para este projeto.

Vale destacar que os dados coletados durante o estudo foram realizados em um referencial fixo, utilizando configurações estáticas no dispositivo. Sendo assim, o uso de Redes Neurais em um futuro trabalho pode se apresentar vantajoso ao lidar com dados coletados em situações variáveis de movimento, como os dados obtidos diretamente no bolso do usuário enquanto pedala. Essa dinâmica de movimento pode exigir modelos mais adaptáveis e complexos, como as Redes Neurais, para lidar com a variabilidade e imprecisão dos dados em tempo real.

Por fim, os resultados alcançados não apenas reforçam a aplicabilidade de aprendizado de máquina no contexto de transporte urbano sustentável, mas também demonstram como tecnologias acessíveis podem ser usadas para gerar impactos positivos na segurança e qualidade de vida dos ciclistas. Este estudo cria margem para futuros trabalhos, como a ampliação do conjunto de dados, com coletas realizadas em diferentes cidades e em contextos geográficos variados, o que pode melhorar a generalização do modelo e torná-lo aplicável a uma maior diversidade de cenários.

Esses resultados também abrem caminho para a integração com outras plataformas, como sistemas de mapeamento, e para o desenvolvimento de soluções de feedback em tempo

real, aprimorando ainda mais a segurança dos ciclistas. O estudo também estabelece um ponto de partida para futuras pesquisas que explorem novas tecnologias e algoritmos, contribuindo para a evolução das soluções de mobilidade urbana sustentável.

REFERÊNCIAS

Abadi, M. et al. (2016). Tensorflow: A system for large-scale machine learning. páginas 265–283.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Alvarez, A. D. et al. (2014). Modeling the driving behavior of electric vehicles using smartphones and neural networks. *IEEE Intelligent Transportation Systems Magazine*, 6(3):44–53.

Basavaraju, A. et al. (2019). A machine learning approach to road surface anomaly assessment using smartphone sensors. *IEEE Sensors Journal*, 20(5):2635–2647.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Chapelle, O. et al., editores (2006). *Semi-Supervised Learning*. The MIT Press.

Chinchor, N. (1992). Muc-4 evaluation metrics. Acesso em 13 jan. 2025.

Eddy, S. R. (1996). Hidden markov models. *Current Opinion in Structural Biology*, 6(3):361–365.

Fazeen, M. et al. (2012). Safe driving using mobile phones. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1462–1468.

François-Lavet, V. et al. (2018). An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*.

Gevorkyan, M. N. et al. (2019). Review and comparative analysis of machine learning libraries for machine learning. *Discrete and Continuous Models and Applied Computational Science*, 27(4):305–315.

Ghahramani, Z. (2003). Unsupervised learning. Em *Summer school on machine learning*, páginas 72–112. Springer Berlin Heidelberg.

Goodfellow, I., Bengio, Y. e Courville, A. (2016). Deep learning. Acesso em 13 jan. 2025.

Google (2023). Visão geral dos sensores. Acesso em 16 nov. 2023.

Jain, A. K., Murty, M. N. e Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31:264–323.

Keras (2024). Sobre o keras 3.

Kotsiantis, S. B. et al. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24.

Lane, N. D. et al. (2010). A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140–150.

LeCun, Y., Bengio, Y. e Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

LeCun, Y. et al. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995.

Loh, W. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

Medsker, L. R. e Jain, L. C. (2001a). Recurrent neural networks: Design and applications. *Design and Applications*, 5:64–67.

Medsker, L. R. e Jain, L. C. (2001b). Recurrent neural networks: Design and applications. Acesso em 13 jan. 2025.

Nasreen, J. et al. (2019). Object detection and narrator for visually impaired people. Em *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, páginas 1–4.

Nelli, F. (2015). *Python Data Analytics: Data Analysis and Science using PANDAs, Matplotlib and the Python Programming Language*. Apress.

Oladipupo, T. (2010). *Types of Machine Learning Algorithms*. InTech.

Paszke, A. et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Powers, D. M. W. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. Acesso em 13 jan. 2025.

Provost, F. e Fawcett, T. (2013). Data science for business: What you need to know about data mining and data-analytic thinking. Acesso em 13 jan. 2025.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. Acesso em 13 jan. 2025.

Sainani, K. L. (2014). Logistic regression. *PMR*, 6:1157–1162.

Shinde, P. P. e Shah, S. (2018). A review of machine learning and deep learning applications. Em *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, páginas 1–6.

Sokolova, M. e Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Acesso em 13 jan. 2025.

StatCounter (2023). Mobile operating system market share worldwide. Acesso em 08 jan. 2024.

Tiwari, S., Bhandari, R. e Raman, B. (2020). Roadcare: A deep-learning based approach to quantifying road surface quality. páginas 231–242.

Weiss, K., Khoshgoftaar, T. M. e Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

Yan, X. e Su, X. G. (2009). *Linear Regression Analysis: Theory and Computing*.