



**UFOP**

Universidade Federal  
de Ouro Preto

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

**Técnicas de Aprendizado de Máquina  
Aplicadas a Classificação da  
Qualidade de Pavimentos Asfálticos  
utilizando Smartphones**

**Samuel Antônio das Neves**

**TRABALHO DE  
CONCLUSÃO DE CURSO**

**ORIENTAÇÃO:**

Vicente José Peixoto de Amorim

**COORIENTAÇÃO:**

Igor Muzetti Pereira

**Fevereiro, 2018  
João Monlevade–MG**

**Samuel Antônio das Neves**

**Técnicas de Aprendizado de Máquina Aplicadas  
a Classificação da Qualidade de Pavimentos  
Asfálticos utilizando Smartphones**

Orientador: Vicente José Peixoto de Amorim

Coorientador: Igor Muzetti Pereira

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

**Fevereiro de 2018**

# FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

## Técnicas de Aprendizado de Máquina Aplicadas a Classificação de Pavimentos Asfálticos utilizando Smartphones

Samuel Antônio das Neves

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI496 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia de Computação e aprovada pela Banca Examinadora abaixo assinada:



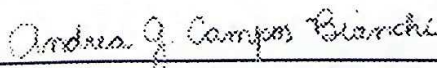
---

Prof. Vicente José Peixoto de Amorim  
Mestre em Ciência da Computação  
DECSI – UFOP



---

Igor Muzetti Pereira  
Mestre em Ciência da Computação  
DECSI – UFOP



---

Andrea Gomes Campos Bianchi  
Doutora em Física de Computacional  
Examinador  
DECOM – UFOP

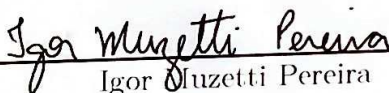
## ATA DE DEFESA

No dia 16 do mês de Fevereiro de 2018, às 09:00 horas, na sala C204 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo(a) aluno(a) **Samuel Antônio das Neves**, sendo a Comissão Examinadora constituída pelos professores: Prof. Vicente José Peixoto de Amorim, Prof. Igor Muzetti Pereira, Prof. Andrea Gomes Campos Bianchi. O(a) candidato(a) apresentou a monografia intitulada: "**Técnicas de Aprendizado de Máquina Aplicadas a Classificação de Pavimentos Asfálticos utilizando Smartphones**". A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 9,8 (nove vírgula oito), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo(a) graduando(a).



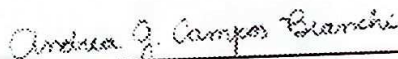
---

Prof. Vicente José Peixoto de Amorim  
Mestre em Ciência da Computação  
DECSI – UFOP



---

Igor Muzetti Pereira  
Mestre em Ciência da Computação  
DECSI – UFOP



---

Andrea Gomes Campos Bianchi  
Doutora em Física de Computacional  
Examinador(a)  
DECOM – UFOP



---

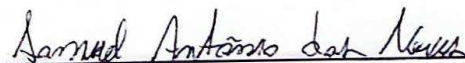
Samuel Antônio das Neves

João Monlevade, 16 de fevereiro de 2018

## TERMO DE RESPONSABILIDADE

Eu, **Samuel Antônio das Neves** declaro que o texto do trabalho de conclusão de curso intitulado “*Técnicas de Aprendizado de Máquina Aplicadas a Classificação de Pavimentos Asfálticos utilizando Smartphones*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 16 de fevereiro de 2018



Samuel Antônio das Neves

*Aos meus pais, irmãos e toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.*

# Agradecimentos

A Deus por ter me dado saúde e força para superar as dificuldades.

A esta universidade, seu corpo docente, direção e administração que propiciaram as condições necessárias para que eu pudesse realizar meus sonhos.

Ao meus orientadores Igor Muzetti e Vicente Amorim, pelo suporte no pouco tempo que lhes coube, pelas suas correções e incentivos.

Aos meus pais, pelo amor, incentivo e apoio incondicional.

A todos membros do laboratório de pesquisa *iMobilis*, pelo companheirismo e ajuda no processo de aprendizagem.

A todos que ajudaram na coleta dos dados utilizados neste trabalho.

E a todos que direta ou indiretamente fizeram parte da minha formação e me apoiaram, o meu muito obrigado.

*“Não se espante com a altura do vôo. Quanto mais alto, mais longe do perigo. Quanto mais você se eleva, mais tempo há de reconhecer uma pane. É quando se está próximo do solo que se deve desconfiar.”*

— Santos Dumont (1873 – 1932)



# Resumo

Os pavimentos rodoviários são de extrema importância para o escoamento de mercadorias, pessoas e serviços. No Brasil cerca de 61,1% do traslado de carga são realizados pelas rodovias (CNT, 2017). Encontrar uma forma de identificar irregularidades nestes pavimentos é de grande importância, pois estudos afirmam que uma má conservação das estradas pode causar impacto na segurança das pessoas que trafegam por elas e até mesmo no consumo de combustível dos veículos (KLAUBERT, 2001). O RoadScan é um sistema desenvolvido por (LIMA et al., 2016) com o intuito de realizar a classificação da qualidade de pavimentos asfálticos e disponibilizá-los em um mapa. Na exibição do mesmo, são utilizadas técnicas de *crowdsourcing* para aprimorar a classificação. Este trabalho desenvolveu uma melhoria para o RoadScan, onde redes neurais foram utilizadas visando aumentar a confiabilidade e a adaptabilidade das classificações de qualidade dos pavimentos. Neste trabalho serão descritos os processos: de extração das características utilizadas na etapa de classificação, de coleta dos dados, de criação da base de dados final e do desenvolvimento de um aplicativo móvel que utiliza redes neurais para classificar a qualidade dos pavimentos. Ao final do trabalho será mostrado um mapa mostrando os resultados alcançados, inclusive em um outro tipo de veículo: motocicletas.

**Palavras-chaves:** aprendizado de máquina, inteligência artificial, qualidade de pavimentos, *smartphones*.

# Abstract

Road pavements are of great importance for the transport of goods, people, and services. In Brazil, about 61.1% of freight is carried out by the highways (CNT, 2017). Finding a way to identify irregularities in these pavements have great importance because studies say that a road in bad conservation can impact the safety of people who travel by them and even in the fuel consumption of vehicles (KLAUBERT, 2001). The RoadScan is a system developed by (LIMA et al., 2016) that classifies the quality of asphalt flooring and make them available on a map. To generate this map, crowdsourcing techniques are used to improve this classification. This work has developed an improvement for the RoadScan, where neural networks have been used aiming the increase of the floor quality classification reliability and adaptability. This work will describe the processes: of features extraction used in the step of classification, data collection, the final database creation and mobile application development that uses neural networks to classify the pavements quality. At the end of the work will be shown a map presenting the results achieved, including another type of vehicle: motorcycles.

**Key-words:** machine learning, artificial intelligence, pavements quality, smartphones.

# Lista de ilustrações

Figura 1 – Comparação entre a venda de dispositivos móveis de diferentes sistemas operacionais. . . . .	20
Figura 2 – Rede Neural <i>multi-layer perceptron</i> (MLP). . . . .	23
Figura 3 – Fluxograma do processo de coleta de dados. . . . .	28
Figura 4 – Interface de coleta de dados. . . . .	29
Figura 5 – Gráfico de amplitude do acelerômetro dos dados coletados em um carro. . . . .	29
Figura 6 – Fluxograma do processo de Análise Primária. . . . .	30
Figura 7 – Arquitetura do sistema. . . . .	33
Figura 8 – Funcionamento do aplicativo no <i>smartphone</i> . . . . .	34
Figura 9 – Tela principal do aplicativo RoadScan. . . . .	35
Figura 10 – Funcionamento do sistema no lado do servidor. . . . .	36
Figura 11 – Mapa classificações geradas pelo RoadScan. . . . .	40

# Lista de tabelas

Tabela 1 – Comparação entre alguns algoritmos classificadores implementados no Weka utilizando a base de dados coletada neste trabalho. . . . .	39
Tabela 2 – Matriz confusão obtida com o algoritmo rede neural MLP no modelo final. . . . .	39
Tabela 3 – Tabela de dados classificação do modelo final. . . . .	40
Tabela 4 – Tabela confusão obtida com o algoritmo SVM implementado pelo <i>Weka</i> . . . . .	48
Tabela 5 – Matriz confusão obtida com o algoritmo rede neural MLP implementado pelo <i>Weka</i> . . . . .	48
Tabela 6 – Matriz confusão obtida com o algoritmo de Naive Bayes implementado pelo <i>Weka</i> . . . . .	48

# Lista de abreviaturas e siglas

**ACM** Association for Computing Machinery

**CNT** Confederação Nacional do Transporte

**CSV** *Comma-separated values*

**FP** Falsos Positivos

**GPS** *Global Positioning System*

**IDE** *Integrated Development Environment*

**IEEE** Institute of Electrical and Electronics Engineers

**KNN** K-Nearest Neighbors

**MLP** *multi-layer perceptron*

**XML** *eXtensible Markup Language*

**SDK** Software Development Kit

**S.O** Sistema Operacional

**SVM** *Support Vector Machine*

**VP** Verdadeiros Positivos

# Lista de símbolos

$\mu$	Letra grega minúscula Mi
$\sigma$	Letra grega minúscula Sigma

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>1.1</b>	<b>O problema de pesquisa</b>	<b>16</b>
<b>1.2</b>	<b>Objetivos</b>	<b>17</b>
1.2.1	Objetivos Gerais	17
1.2.2	Objetivos Específicos	17
<b>1.3</b>	<b>Organização do trabalho</b>	<b>17</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>18</b>
<b>2.1</b>	<b>Trabalhos Correlatos</b>	<b>18</b>
<b>2.2</b>	<b>Conceitos Básicos</b>	<b>20</b>
2.2.1	Android	20
2.2.2	Aprendizado de Máquina	21
2.2.2.1	Aprendizado Supervisionado	21
2.2.2.1.1	Redes Neurais	21
2.2.2.2	Aprendizado Não-Supervisionado	23
2.2.2.3	Ruído no aprendizado de máquina	23
2.2.3	Softmax	24
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>25</b>
<b>3.1</b>	<b>Materiais e Ferramentas</b>	<b>25</b>
3.1.1	Weka	25
3.1.2	Python	25
3.1.3	TensorFlow	26
<b>3.2</b>	<b>Linguagens de Programação</b>	<b>26</b>
<b>3.3</b>	<b>Dados Coletados</b>	<b>26</b>
3.3.1	Coleta	27
3.3.2	Análise primária	28
3.3.2.1	Padronização	31
3.3.2.2	características Utilizadas	31
3.3.3	Análise secundária	32
3.3.4	Classes utilizadas na classificação	32
<b>3.4</b>	<b>Funcionamento do sistema</b>	<b>32</b>
3.4.1	Smartphone	32
3.4.2	Servidor	36
3.4.2.1	Adição de pavimentos na base de dados do servidor	37
3.4.2.2	Requisições do mapa	37

3.4.3	Visualização do Mapa por Usuários Finais . . . . .	37
<b>4</b>	<b>RESULTADOS . . . . .</b>	<b>38</b>
4.1	Resultados Experimentais . . . . .	38
4.2	Resultados Finais . . . . .	40
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>41</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>42</b>
	<b>ANEXOS . . . . .</b>	<b>45</b>
	<b>ANEXO A – FEATURES UTILIZADAS . . . . .</b>	<b>46</b>
A.1	Média . . . . .	46
A.2	Desvio Padrão . . . . .	46
A.3	Coefficientes de variação . . . . .	46
A.4	Diferença pico a pico . . . . .	46
A.5	Obliquidade(Skewness) . . . . .	46
A.6	Curtose(Kurtosis) . . . . .	47
A.7	Energia . . . . .	47
A.8	Potência . . . . .	47
A.9	Raiz do valor quadrático médio . . . . .	47
A.10	Velocidade . . . . .	47
	<b>ANEXO B – MATRIZ DE CONFUSÃO DOS ALGORITMOS TESTADOS NO WEKA . . . . .</b>	<b>48</b>



# 1 Introdução

## 1.1 O problema de pesquisa

O transporte rodoviário é de grande importância para o deslocamento de produtos e pessoas, de acordo com a Confederação Nacional de Transportes (CNT, 2017) cerca de 61,1% do traslado de carga no Brasil se dá através das rodovias. Ainda segundo a CNT 58% das rodovias brasileiras possuem algum problema, e 48,3% dos trechos avaliados são classificados como regulares, ruins ou péssimos.

Uma série de fatores podem influenciar na deterioração dos pavimentos asfálticos, como: o tráfego de veículos, a umidade, solo e a qualidade da construção do pavimento (GEORGE; RAJAGOPAL, 2015). Estudos afirmam que quando em estado de conservação precário, os mesmos podem causar efeitos indesejáveis como o aumento do consumo de combustível dos veículos (KLAUBERT, 2001), além disso pode-se inferir que pavimentos ruins impactam na segurança e na velocidade do fluxo dos veículo. Tendo em vista a importância dos pavimentos asfálticos surge uma grande necessidade de se desenvolver uma solução que possibilite a detecção de trechos com irregularidades e a classificação da qualidade dos mesmos. Um padrão internacional chegou a ser criado, visando determinar uma métrica confiável para classificar a qualidade de pavimentos (SAYERS, 1986). Entretanto, a principal dificuldade encontrada na ampla utilização deste índice de classificação é sua complexidade de implementação e custo computacional, visto que o mesmo utiliza cálculos avançados.

Segundo uma pesquisa realizada em 2017 pela Fundação Getúlio Vargas, no Brasil tem-se em média um *smartphone* por habitante, são cerca de 280 milhões de dispositivos sendo utilizados para os mais variados fins (MEIRELLES, 2017). O baixo custo e a popularização dos *smartphones* servem como motivação para o desenvolvimento de aplicações que consigam aproveitar os sensores e funcionalidades presentes nestes dispositivos.

Em 2008 foi proposto o Nericell (MOHAN; PADMANABHAN; RAMJEE, 2008): um sistema para o monitoramento do tráfego e das estradas. O Nericell foi o pioneiro no uso dos *smartphones* para este fim. Ele utiliza sensores como *Global Positioning System* (GPS), microfone e acelerômetro, e baseado na leitura e análise dos dados lidos consegue fazer a detecção de buracos e freatas. Diversos trabalhos foram desenvolvidos posteriormente com o objetivo de detectar a condição dos pavimentos através de *smartphones* (CHUGH; BANSAL; SOFAT, 2014).

Aqui, será apresentada uma solução que utiliza *smatphones* em conjunto com redes neurais para realizar a classificação da qualidade dos pavimentos, exibindo-a em um tempo

de resposta aceitável para o usuário e ainda disponibilizando estes dados posteriormente em um *site* para visualização pública.

## 1.2 Objetivos

### 1.2.1 Objetivos Gerais

O objetivo deste trabalho é criar um sistema que seja capaz de utilizar técnicas de aprendizado de máquina e *smartphones* para realizar a classificação da qualidade de pavimentos asfálticos.

### 1.2.2 Objetivos Específicos

Abaixo são citados os objetivos específicos que este trabalho visa alcançar para atingir o objetivo geral.

- Implementar uma forma de se realizar a coleta de dados do acelerômetro presente no *smartphone*.
- Extrair característica dos dados coletados que consigam representar os pavimentos.
- Criar uma base de dados para servir como entrada de algoritmos de aprendizado supervisionado.
- Pesquisar algoritmos de aprendizado de máquina que possam ser aplicados na classificação dos pavimentos baseados na leitura do acelerômetro.
- Treinar uma rede neural para reconhecer a qualidade dos pavimentos asfálticos.
- Desenvolver um aplicativo que utilize o conhecimento obtido pelos algoritmos de aprendizagem para realizar a classificação dos pavimentos no *smartphone*.

## 1.3 Organização do trabalho

O restante deste trabalho é organizado como segue. O Capítulo 2 apresenta os trabalhos correlatos e contextualiza alguns conceitos básicos necessários para a compreensão deste. As informações sobre o desenvolvimento deste trabalho estão descritas no Capítulo 3. No Capítulo 4 são discutidos os resultados alcançados pelo sistema. Por fim no Capítulo 5 são abordadas as conclusões, desafios encontrados e perspectivas futuras.

## 2 Revisão bibliográfica

Neste capítulo serão abordados alguns trabalhos correlatos à pesquisa, em seguida diversos conceitos necessários para o entendimento do trabalho desenvolvido.

### 2.1 Trabalhos Correlatos

Como forma de relacionar estes trabalhos correlatos, buscou-se em *sites* como Institute of Electrical and Electronics Engineers ([IEEE](http://ieeexplore.ieee.org/Xplore/home.jsp))<sup>1</sup> e Association for Computing Machinery ([ACM](https://dl.acm.org/))<sup>2</sup> artigos usando as seguintes palavras chaves: *machine learning, pavements, quality, road health, classification, smartphones, accelerometer, features extraction*.

Diversas publicações foram realizadas para estudar formas eficientes de classificar a qualidade de pavimentos asfálticos, surgiram inclusive *surveys* ([CHUGH; BANSAL; SOFAT, 2014](#)) analisando diversos artigos sobre este tema. Para embasar este trabalho cinco foram escolhidas. Todos os trabalhos citados abaixo utilizam o sinal do acelerômetro como base para realizar análise dos pavimentos. Abaixo uma relação de publicações que apresentam uma correlação com este trabalho:

1. **RoadScan:** Trabalho desenvolvido em 2016 na Universidade Federal de Ouro Preto pelo laboratório de pesquisa Imobilis. O trabalho desenvolvido utilizou o sinal coletado pelo sensor acelerômetro presente nos *smartphones*, em conjunto com dados de geolocalização. A estratégia abordada consiste em através da medida do desvio padrão dos dados de acelerômetro coletados, realizar uma classificação utilizando alguns limiares descobertos empiricamente. Os resultados obtidos foram satisfatórios, porém o método de classificação utiliza limiares fixos de difícil ajuste, o que dificulta a adaptação das classificações realizadas para diferentes necessidades. ([LIMA et al., 2016](#)).
2. **RoADS:** No trabalho ([SERAJ et al., 2016](#)) são utilizadas técnicas avançadas para a extração das características dos dados de acelerômetro. Após selecionadas as características, o classificador [SVM\(CORTES; VAPNIK, 1995\)](#) é utilizado para realizar a identificação de anomalias nos pavimentos asfálticos. O autor busca utilizar *smartphones* em seu trabalho, devido ao baixo custo destes dispositivos. Todo o tratamento dos dados coletados é feito em um dispositivo externo. O pré-processamento dos dados para a extração das características garantiram resultados muito bons, chegando a alcançar uma acurácia de 91% na detecção destas anomalias.

<sup>1</sup> <<http://ieeexplore.ieee.org/Xplore/home.jsp>>

<sup>2</sup> <<https://dl.acm.org/>>

Porém, dada a complexidade desta tarefa, é improvável que o sistema possa ser implementado com um tempo de resposta aceitável utilizando *smartphones* para o processamento.

3. Em (MOHAMED et al., 2016), o autor realiza o mapeamento de anomalias nos pavimentos de forma similar ao trabalho citado anteriormente, porém, o mesmo utiliza o algoritmo *perceptron* (ROSENBLATT, 1958) para realizar a classificação destas imperfeições. Como entrada do algoritmo *perceptron* ele utiliza a leitura do eixo Z do acelerômetro perpendicular à superfície da Terra. Este trabalho trata apenas a detecção de anomalias, não trata a qualidade do pavimento no geral.
4. Em (BADUROWICZ; CIEPLAK; MONTUSIEWICZ, 2016), os autores criaram um sistema onde dados coletados pelo acelerômetro de *smartphones* são enviados para um servidor. O mesmo analisa os dados recebidos e gera um valor para aquele trecho de estrada, retornando-o aos *smartphones* dos usuários participantes deste sistema social. Como trabalho futuro, os autores pretendem aprimorar o cálculo deste valor gerado, para que possa ser utilizado para classificar a qualidade de um pavimento. Um possível empecilho para que o sistema desenvolvido possa ser utilizado em países com um índice menor de desenvolvimento é o fato de que a qualidade da internet móvel disponibilizada nos mesmos tende a ser inferior, prejudicando assim o tempo de resposta ao usuário.
5. Em (WANG; CHUANG; YI, 2016), um sistema *crowdsourcing* é criado para identificação de anomalias em pavimentos. A arquitetura do mesmo é similar ao deste trabalho. Porém, o mesmo utiliza modelos matemáticos para classificação de anomalias ao invés de algoritmos de inteligência artificial. Com o método utilizado, Wang consegue alcançar 88% de precisão na detecção de anomalias seguras e 82% na detecção de anomalias consideradas perigosas. Apesar dos bons resultados, este trabalho busca a classificar o nível de perigo das irregularidades encontradas em pavimentos, e não classificar a qualidade dos pavimentos.

Os trabalhos citados acima mostram que é possível obter bons resultados utilizando *smartphones* para detectar anomalias e classificar a qualidade dos pavimentos. Eles contribuíram com este trabalho com informações sobre quais características utilizar e estratégias para criar a base de dados. O sistema desenvolvido aqui consiste em utilizar técnicas de aprendizado de máquina para classificar a qualidade dos pavimentos unindo conceitos e ideias abordados nestes trabalhos correlatos.

Figura 1 – Comparação entre a venda de dispositivos móveis de diferentes sistemas operacionais.



Fonte: (GARTNER, 2017)

## 2.2 Conceitos Básicos

Será apresentada nesta seção alguns conceitos sobre Android e aprendizado de máquina.

### 2.2.1 Android

O sistema operacional é um programa responsável por agir como uma camada intermediária entre o *hardware* e o usuário. Para cumprir este objetivo o mesmo oferece serviços tanto para os desenvolvedores quanto para os usuários finais, de forma com que estes não necessitem empregar comandos à nível de *hardware*(STALLINGS, 2008). Na categoria de dispositivos móveis, 3 sistemas operacionais se destacam: Android, iOS, Windows Phone.

Sendo baseado no *kernel* do Linux, o Android é um sistema *open source*. Isto permite com que o mesmo seja customizado de diversas formas, possibilitando com que empresas que desenvolvam dispositivos e *smartphones* possam adaptar o sistema às suas necessidades.

Na Figura 1 pode-se perceber o grande impacto dos dispositivos Android no

mercado de dispositivos móveis. Nota-se que com uma considerável margem de vantagem a venda de dispositivos que utilizam Android supera a dos outros sistemas operacionais como iOS e Windows Phone. Estima-se que os dispositivos Android ocupem cerca de 70% do mercado mundial (NOVAC et al., 2017). É esperado que o número de usuários deste Sistema Operacional (S.O) se mantenha alto nos próximos anos graças a este alto número de vendas nos últimos anos. Devido a esta popularidade escolheu-se esta plataforma como hospedeira para o sistema móvel desenvolvido.

## 2.2.2 Aprendizado de Máquina

O aprendizado de máquina é a área da inteligência artificial responsável pelos métodos e algoritmos que possuem a capacidade de aprender com informações obtidas através de uma base de dados. O conceito desta forma de cognição difere do processo cognitivo humano, porém, se baseia no mesmo. Parte do princípio de que através do erro se consegue melhorar uma dada resposta, ou seja, consiste em acreditar que o aprendizado é adquirido através da experiência. Os algoritmos desta área podem ser divididos em diferentes categorias, dentre elas pode-se citar o aprendizado supervisionado que foi utilizado neste trabalho e o não supervisionado.

### 2.2.2.1 Aprendizado Supervisionado

O aprendizado supervisionado consiste no conjunto de técnicas em que existe um “supervisor” que sabe previamente qual é a resposta esperada de um sistema dado uma entrada. Para efetuar o aprendizado, o algoritmo compara sua saída com a deste supervisor, para que assim, se ajuste em busca de minimizar o erro de sua resposta (BISHOP, 2006).

De forma simplificada: a base de dados utilizada nestas técnicas possui as denominadas *flags* ou marcadores, que representam a resposta esperada. O sistema então se molda, fazendo com que dado uma entrada, seja emitida como saída o mesmo valor do supervisor.

Existem diversos algoritmos que seguem esta estratégia, como por exemplo: *Support Vector Machine* (SVM) (CORTES; VAPNIK, 1995), Regressão Linear e Redes Neurais, as quais foram utilizadas neste trabalho.

#### 2.2.2.1.1 Redes Neurais

O termo “redes neurais” teve sua primeira aparição no artigo “*A Logical Calculus of Ideas Immanent in Nervous Activity*” de (MCCULLOCH; PITTS, 1943) onde pela primeira vez, de uma forma metodológica e científica tentou se descrever matematicamente como acontece o processamento de uma informação em um sistema biológico. Desde então este termo vem sendo utilizado por vários modelos de aprendizado de máquina, onde

alguns deles sofrem até mesmo críticas em relação a sua correspondência com a biologia (BISHOP, 2006).

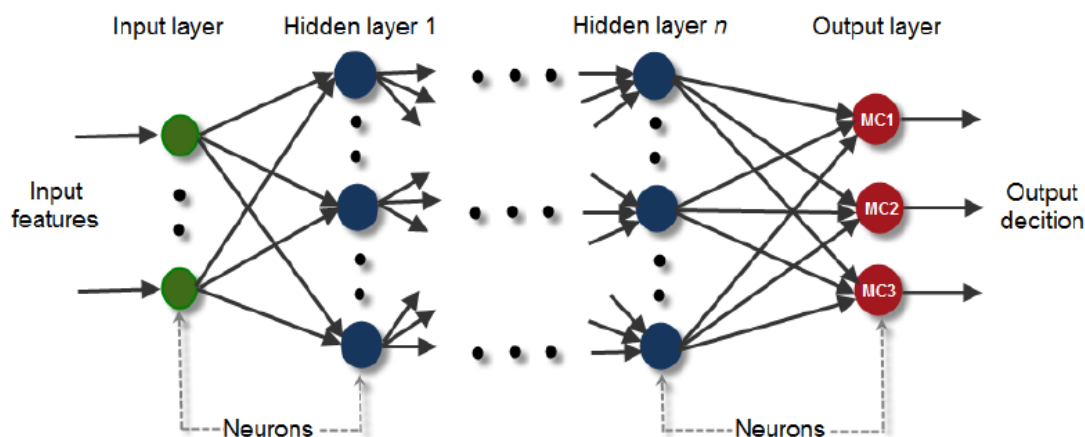
O trabalho de McCulloch e Pitts se concentrou mais no processo de descrever um modelo artificial de um neurônio e representar suas capacidades computacionais do que em apresentar uma técnica de aprendizado em si. Em 1958 Frank Rosenblatt, um psicólogo americano, desenvolveu um modelo promissor denominado *perceptron*, que simula o processo de reconhecimento de padrões através de uma estrutura que representa um neurônio. Esta estrutura era constituída de 3 camadas: uma camada inicial responsável por receber as entradas de um meio externo, uma camada intermediária de pesos ajustáveis e por fim uma camada de saída, que possui a resposta do modelo (BRAGA; CARVALHO; LUDERMIR, 2007).

O *perceptron* surgiu como uma grande promessa no ramo da inteligência artificial, contudo, o mesmo possui a grande limitação de conseguir distinguir apenas padrões que sejam linearmente separáveis conforme apresentado por (MINSKY; PAPERT, 1967). Este estudo congelou os ânimos em relação ao uso das redes neurais artificiais na classificação de padrões. Porém, após a descrição do algoritmo *backpropagation* surgir, provou-se que as redes neurais podem sim resolver os problemas ditos não-linearmente separáveis (RUMELHART; HINTON; WILLIAMS, 1986). A partir do surgimento deste algoritmo, tornou-se viável novamente a utilização de redes constituídas de diversos neurônios interligados, similares ao do modelo proposto por McCulloch e Pitts.

Um exemplo de rede neural dita de múltiplas camadas ou *multi-layer perceptron* pode ser visto na Figura 2. Neste novo modelo nota-se agora a existência de mais camadas e neurônios, onde até a resposta de cada neurônio da camada anterior serve como entrada para as posteriores. Pode-se perceber também através da figura uma nova camada denominada oculta ou (*hidden layer*), onde uma rede neural pode conter várias camadas ocultas.

De forma breve, neste modelo tem-se que os dados de entrada são inseridos na camada de entrada e em seguida são multiplicados pelos pesos aleatoriamente gerados. A saída de cada neurônio da camada de entrada pode ser descrita pela Equação 2.1, onde  $Y$  é a saída do neurônio  $k$ ,  $X$  as  $n$  entradas e  $W$  os  $n$  pesos atribuídos. Antes de avançar, estas saídas  $Y$  passam por uma espécie de filtro denominada função de ativação. As funções de ativação geralmente são funções não lineares, cujo objetivo é possibilitar com que as saídas da rede neural possam ser valores contínuos diferentes de 0 e 1. Após passar por esta função de ativação, a resposta de um neurônio da camada anterior passa a ser a entrada do neurônio da camada posterior. Este processo se repete até que se alcance a última camada onde os neurônios são modelados de forma com que se obtenha a resposta do sistema no formato desejado. Nesta camada costumam-se utilizar funções específicas para avaliar o resultado, como o caso da função *softmax*, que foi utilizada neste trabalho e será abordada ainda neste capítulo.

Figura 2 – Rede Neural MLP.



Fonte: (EBRAHIMI; MOLLAZADE; AREFI, 2012)

Esta breve descrição relata o funcionamento de uma iteração do processo de aprendizado de uma rede neural para um exemplo, porém, para que a rede neural consiga representar um modelo eficiente são necessárias várias iterações e vários exemplos.

O aprendizado de uma rede neural está contido nos pesos de cada neurônio, ao final de cada iteração estes pesos são ajustados através do algoritmo *backpropagation*, onde o erro obtido através da camada de saída se propaga por todas as camadas anteriores fazendo com que os pesos de cada neurônio possam ser ponderados baseados neste erro.

$$Y_k = \sum_{i=1}^n W_{ki} * X_{ki} \quad (2.1)$$

#### 2.2.2.2 Aprendizado Não-Supervisionado

No aprendizado supervisionado já se conhece previamente a saída esperada e através dela se aprimora o conhecimento. Em contrapartida, no aprendizado não supervisionado o algoritmo atua sem um supervisor. Nestes algoritmos não existe influência humana direta e o conhecimento se dá a partir do agrupamento dos dados baseados em suas similaridades. Nesta classe de algoritmos temos por exemplo, o K-Means e o K-Nearest Neighbors (KNN).

#### 2.2.2.3 Ruído no aprendizado de máquina

O grande objetivo dos ditos algoritmos de aprendizagem indutivos é, dado um conjunto de instâncias de treinamento, criar um sistema que maximize a precisão alcançada nas classificações, minimizando conseqüentemente o erro.

Para alcançar bons resultados, é importante que a base de dados utilizada possua uma boa qualidade. O ruído é um dos fatores que mais interfere. Ele pode ser dividido em dois tipos, o de atributos e o de classes (WU, 1995). O ruído dos atributos é geralmente



causado por erros na coleta dos dados, isso causa os chamados *outliers* que podem ser descritos como uma amostra que possui dados discrepantes se comparada à sua população. Os ruídos de classes podem refletir dois problemas: exemplos contraditórios e instâncias classificadas incorretamente (ZHU; WU, 2004).

### 2.2.3 Softmax

A função Softmax é geralmente utilizada na última camada de redes neurais para que a saída seja obrigatoriamente de uma das classes definidas. Ela consiste em ponderar a probabilidade de um evento de uma população ocorrer com base no número de ocorrências. De maneira prática ela atribui uma probabilidade a cada neurônio da camada de saída, e cada um destes representa uma classe. Assim a classe escolhida é aquela com maior probabilidade. A função Softmax pode ser observada na Equação 2.2, onde a variável  $y$  corresponde às respostas obtidas na camada de saída da rede neural.

$$P(y_i) = \frac{e^{y_i - \max(y)}}{\sum_{i=1}^n e^{y_i - \max(y)}} \quad (2.2)$$

## 3 Desenvolvimento

Este capítulo descreve brevemente os materiais e ferramentas utilizados no RoadScan, além de apresentar o processo executado para a classificação dos pavimentos.

### 3.1 Materiais e Ferramentas

Nesta seção serão descritas algumas ferramentas e materiais que foram utilizados durante a fase de desenvolvimento.

#### 3.1.1 Weka

O Weka é um *framework* que contém uma coleção de algoritmos de aprendizado de máquina, desenvolvido pela Universidade de Waikato na Nova Zelândia, ele possibilita com que se consiga testar uma base de dados utilizando diversos algoritmos da literatura. Além disto, o Weka é um *software* de código aberto o que facilita o entendimento do funcionamento de alguns dos métodos por ele implementados([WEKA, 2018](#)).

#### 3.1.2 Python

A linguagem de programação Python foi concebida no final dos anos 80 e sua implementação começou em dezembro de 1989 por Guido van Rossum. Python é atualmente uma das mais populares linguagens dinamicamente tipadas de propósito geral, junto com Perl, PHP, Ruby. Ela pode ser utilizada em diversas aplicações desde *web servers* que proveem serviços 24 horas por dia, sistemas com interface gráfica, testes automatizados, entre outros. É utilizada tanto por cientistas nos computadores mais rápidos do mundo quanto por crianças aprendendo a programar([ROSSUM, 2009](#)).

Uma das áreas onde a linguagem Python vem se destacando recentemente é em aplicações que utilizam aprendizado de máquina e *data science*, pois, apesar de ser uma linguagem interpretada possui um desempenho considerável quando trabalha com um grande número de dados, ademais é de fácil compreensão/aprendizado e existem inúmeras bibliotecas que facilitam a utilização de alguns algoritmos consagrados da área.

Dentre as bibliotecas disponíveis, pode-se citar o TensorFlow ([TENSORFLOW, 2018](#)), uma biblioteca multi linguagem utilizada neste trabalho e que será introduzida posteriormente. Além desta, outras bibliotecas como NumPy, Matplotlib e Pandas foram utilizadas para diversos fins, entre eles, o uso de algumas funções matemáticas, a visualização gráfica dos dados e representação dos mesmos.

### 3.1.3 TensorFlow

TensorFlow é um *software* matemático e de código aberto desenvolvido em 2011 pelo Google Brain Team. Seu objetivo inicial é conduzir pesquisas no ramo de aprendizado de máquina e redes neurais profundas em que o aprendizado acontece nas camadas ocultas. Porém, o mesmo é genérico o suficiente a ponto de ser aplicado em uma grande variedade de soluções(ZACCONE, 2017).

## 3.2 Linguagens de Programação

Para o desenvolvimento da aplicação móvel foi utilizado o Android Software Development Kit (SDK) em conjunto com a linguagem Java para elaboração da parte lógica, e *eXtensible Markup Language* (XML) para definição da interface gráfica. Como *Integrated Development Environment* (IDE), utilizou-se o Android Studio<sup>1</sup> para auxiliar no processo de desenvolvimento da aplicação. Escolheu-se trabalhar com esta plataforma devido a grande popularidade dos dispositivos, o que facilitou a procura por conteúdo sobre desenvolvimento e acesso aos dispositivos para testes.

Além da linguagem Java utilizada para o desenvolvimento do aplicativo, utilizou-se Python em conjunto com algumas bibliotecas<sup>2</sup> entre elas o TensorFlow. Ela foi utilizada para desenvolver a rede neural responsável pela classificação. A rede neural desenvolvida se trata de uma rede *multi-layer perceptron* (MLP) citada na Seção 2.2.2.1.1. Esta rede utiliza o algoritmo *backpropagation* para ajuste dos pesos, e o Softmax na camada de saída. Assim, na camada de saída tem-se uma distribuição de probabilidade, em que a classe que representa a classificação é aquela com maior valor.

Para complementar, durante o desenvolvimento do *site* foram utilizadas linguagens como PHP<sup>3</sup>, JavaScript<sup>4</sup>, jQuery<sup>5</sup>. Elas tiveram como objetivo realizar a comunicação com o banco de dados e efetuar a apresentação dos pavimentos classificados ao usuário.

Por fim, como sistema de gerenciamento de banco de dados optou-se por manter o MySQL<sup>6</sup> devido a conhecimentos prévios com este sistema.

## 3.3 Dados Coletados

Nesta seção será descrito o processo de coleta e análise dos dados utilizados para treinar a rede neural.

<sup>1</sup> <<https://developer.android.com/studio/index.html>>

<sup>2</sup> Matplotlib, NumPy, Pandas.

<sup>3</sup> PHP- <<http://www.php.net/>>

<sup>4</sup> JavaScript- <<https://www.javascript.com/>>

<sup>5</sup> jQuery- <<https://jquery.com/>>

<sup>6</sup> MySQL<<https://www.mysql.com/>>

### 3.3.1 Coleta

Para realizar a coleta dos dados desenvolveu-se inicialmente um aplicativo paralelo ao RoadScan, denominado “RSTrainer”. Sua interface é simples e possui um total de seis botões, sendo dois deles utilizados para iniciar e parar a coleta. Os outros quatro permitem que se escolha a classificação dos pavimentos. O aplicativo realiza a coleta dos dados do acelerômetro dividindo-os em pavimentos de 5 metros e atribuindo aos mesmos a classificação selecionada pelos botões.

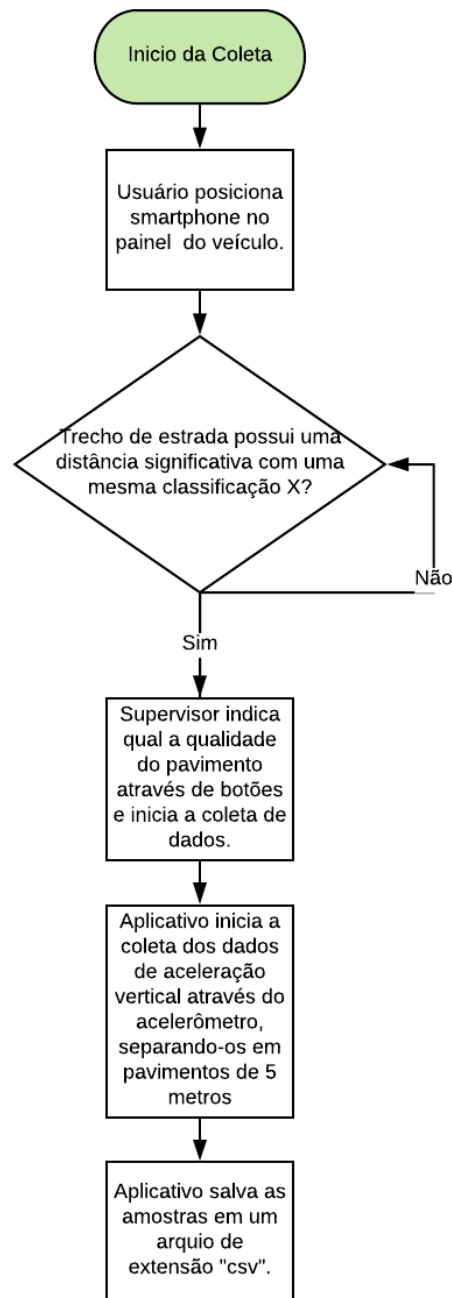
Inicialmente os dados foram armazenados em um banco de dados relacional nativo dos dispositivos Android: o SQLite, porém a frequência de coleta de dados provenientes do acelerômetro é muito alta e a inserção das leituras neste banco se tornou um gargalo pois a gravação não conseguia acompanhar a leitura. Assim a escrita destes dados em um arquivo de extensão *Comma-separated values* (CSV) tornou-se mais viável por ser mais rápida, e os dados utilizados na classificação passaram a ser salvos na memória interna do dispositivo.

Após um tempo o RSTrainer foi incorporado ao aplicativo RoadScan sofrendo apenas algumas modificações (Figura 4).

A coleta dos dados utilizados neste trabalho foi feita em carros e motos, onde se trafegou por trechos específicos de forma controlada, visando realizar a coleta dos dados de todos os quatro tipos de pavimentos. No caso de carros esta coleta foi realizada pelo passageiro, para não atrapalhar a concentração do motorista. A coleta dos dados em uma motocicleta foi realizada da seguinte forma: em um local pré-estabelecido ligou-se o aplicativo e iniciou-se o percurso, ao final do percurso o motociclista imobilizava o veículo e parava a coleta dos dados.

Todos estes procedimentos foram realizados sem desviar a atenção do motorista/piloto do trânsito. A Figura 3 mostra um fluxograma que representa este processo de coleta de dados nos smartphones. Ao final deste processo é gerado uma base de dados crus, que possui leituras de acelerômetro separadas a cada pavimento de 5 metros. O arquivo foi utilizado em um computador na fase denominada aqui de “Análise Primária”.

Figura 3 – Fluxograma do processo de coleta de dados.



Fonte: Elaborado pelo autor.

### 3.3.2 Análise primária

A análise dos dados e extração das características é realizada utilizando a linguagem Python juntamente com algumas bibliotecas auxiliares. De início, realizou-se a plotagem dos dados para verificar se o gráfico da amplitude se comportava como o esperado. Este gráfico pode ser visualizado na Figura 5, onde são representados os dados de aceleração do eixo Z coletados em um carro. Além disto é possível observar o que já era esperado: os

Figura 4 – Interface de coleta de dados.



(a) RSTrainer

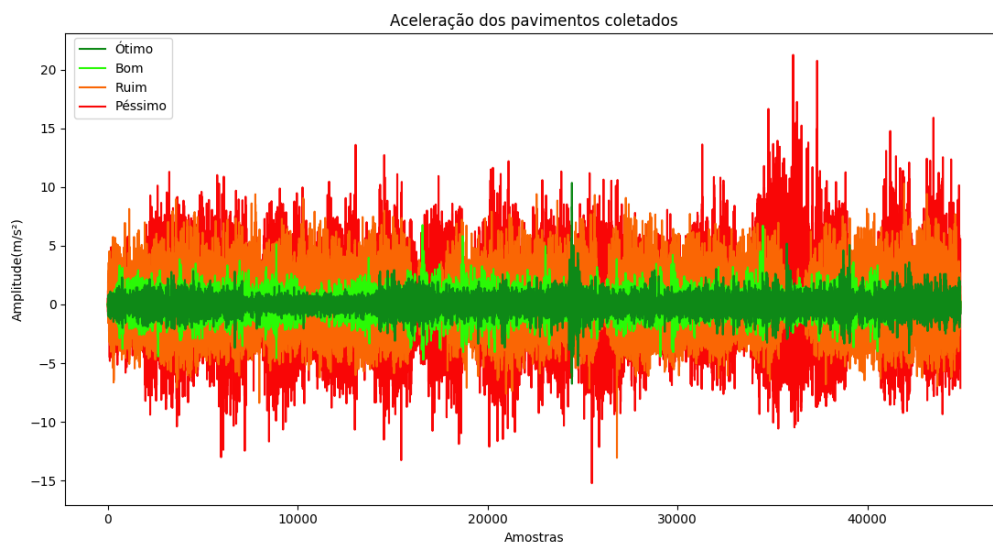


(b) RSTrainer após incorporação ao RoadScan.

Fonte: Elaborado pelo autor.

pavimentos classificados como horríveis possuem uma amplitude maior do que as outras classificações, e conforme a qualidade do pavimento aumenta esta amplitude tende à diminuir. É perceptível ainda que existem dados com ruídos, onde mesmo os trechos classificados como bons possuem picos de amplitude.

Figura 5 – Gráfico de amplitude do acelerômetro dos dados coletados em um carro.



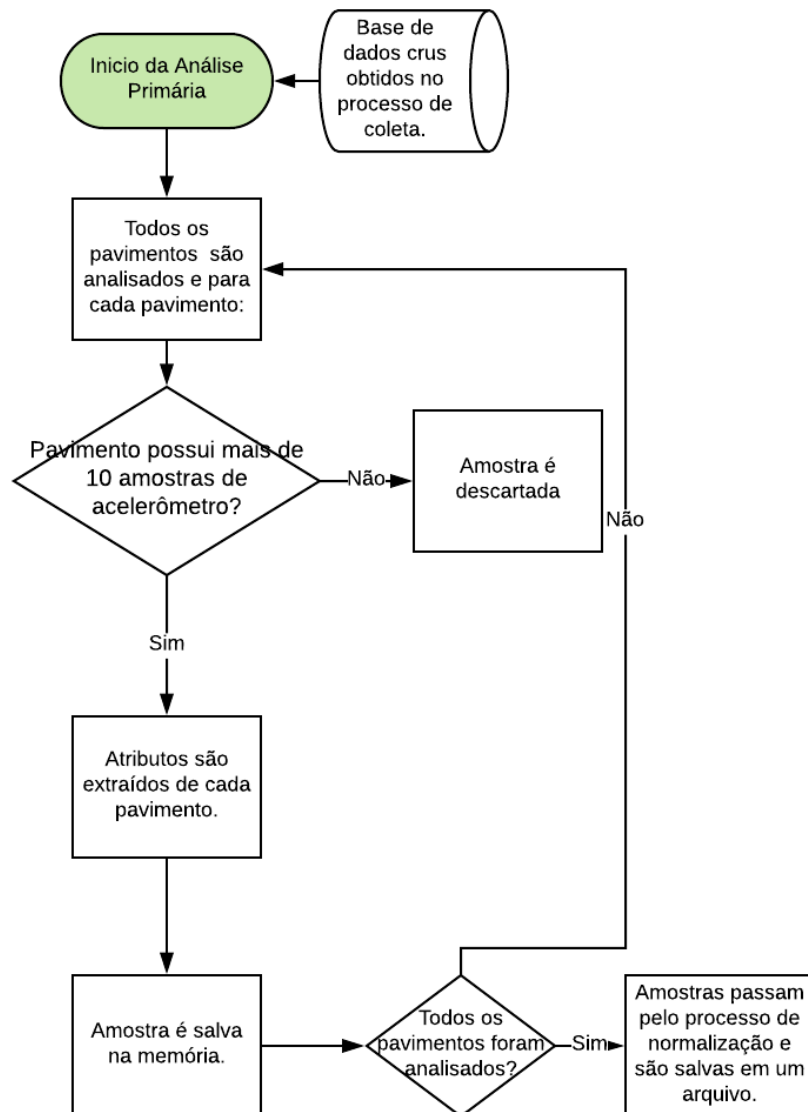
Fonte: Elaborado pelo autor.

Estes ruídos foram causados durante o processo de classificação e coleta dos dados,

devido a dificuldade de se encontrar trechos contínuos que possam representar uma das classificações. No próximo passo realizou-se a extração das características apresentadas na Seção 3.3.2.2.

Por fim, foi efetuada a padronização dos dados de acordo com o método descrito na Seção 3.3.2.1. Este processo de padronização gerou certos valores constantes que foram salvos em um arquivo, e são utilizados no processo final de classificação, realizado no aplicativo móvel. Todo o processo pode ser simplificado através fluxograma da Figura 6.

Figura 6 – Fluxograma do processo de Análise Primária.



Fonte: Elaborado pelo autor.

A nova base de dados já padronizada foi salva em um novo arquivo de extensão CSV. Este arquivo foi utilizado em uma nova fase deste trabalho denominada Análise Secundária dos dados, descrita na Seção 3.3.3.

### 3.3.2.1 Padronização

A padronização/normalização é um recurso muito utilizado na estatística que consiste em subtrair a média de um valor pertencente a um conjunto de dados e dividir o resultado pelo desvio padrão. No trabalho as características são normalizadas após serem extraídas do vetor de pavimentos. A padronização é feita através da equação 3.1, onde  $X$  representa o valor,  $\mu_x$  a média dos valores de  $X$  e  $\sigma_x$  o desvio padrão. Após feita esta normalização, tem-se que a média dos novos dados ficará centrada em 0 e seu desvio padrão em 1, o que em redes neurais diminui o efeito de polarização (GEMAN; BIENENSTOCK; DOURSAT, 1992), efeito este responsável por afetar a precisão do modelo.

$$X_{new} = \frac{X - \mu_x}{\sigma_x} \quad (3.1)$$

### 3.3.2.2 características Utilizadas

As características ou *features* são atributos extraídos de uma base de dados brutos. Após extrair estes atributos é criada uma nova base de dados que serve de entrada para algoritmos de aprendizado de máquina. A escolha das características impacta muito na assertividade do modelo gerado pela rede neural, bem como no desempenho do aplicativo móvel tendo em vista que as mesmas serão extraídas também no aplicativo. No princípio um conjunto grande de características foram levantadas baseadas nos artigos: (SERAJ et al., 2016) e (CASALE; PUJOL; RADEVA, 2011). Porém, algumas dessas características são complexas de serem calculadas computacionalmente, como por exemplo as que são extraídas do domínio da frequência. Isto impactaria negativamente no desempenho final da aplicação, e não agregaria uma maior precisão no modelo, logo foram descartadas. Na versão final optou-se pelas características descritas abaixo:

- Média
- Desvio Padrão
- Coeficiente de variação
- Diferença pico a pico
- Obliquidade
- Curtose
- Energia
- Potência
- Raiz do valor quadrático médio



- Velocidade

Uma melhor descrição de cada uma destas características pode ser visualizada no anexo [A](#).

### 3.3.3 Análise secundária

No processo de análise secundária, a base de dados criada no processo anterior foi testada utilizando o *software* Weka introduzido na Seção [3.1.1](#). Ele facilitou muito o processo de testes, devido às estatísticas geradas pelo mesmo e sua interface gráfica de fácil usabilidade. Através dele foi possível testar a base de dados com diversos algoritmos de aprendizado de máquina, possibilitando aferir a precisão de cada um dos modelos gerados.

Após estes testes, a base de dados foi utilizada como entrada para uma rede neural *multi-layer perceptron* (MLP) implementada em Python. A rede foi desenvolvida utilizando a biblioteca TensorFlow e salva em um arquivo de extensão “.pb”. Este arquivo pode ser integrado através de uma versão da mesma biblioteca em uma aplicação Android.

### 3.3.4 Classes utilizadas na classificação

Como dito anteriormente, os pavimentos são classificados em quatro níveis de qualidade distintos. A cada um deles é atribuída uma cor correspondente:

- Pavimento Ótimo: representado pela cor verde escuro.
- Pavimento Bom: representado pela cor verde claro.
- Pavimento Ruim: representado pela cor laranja.
- Pavimento Péssimo: representado pela cor vermelho.

## 3.4 Funcionamento do sistema

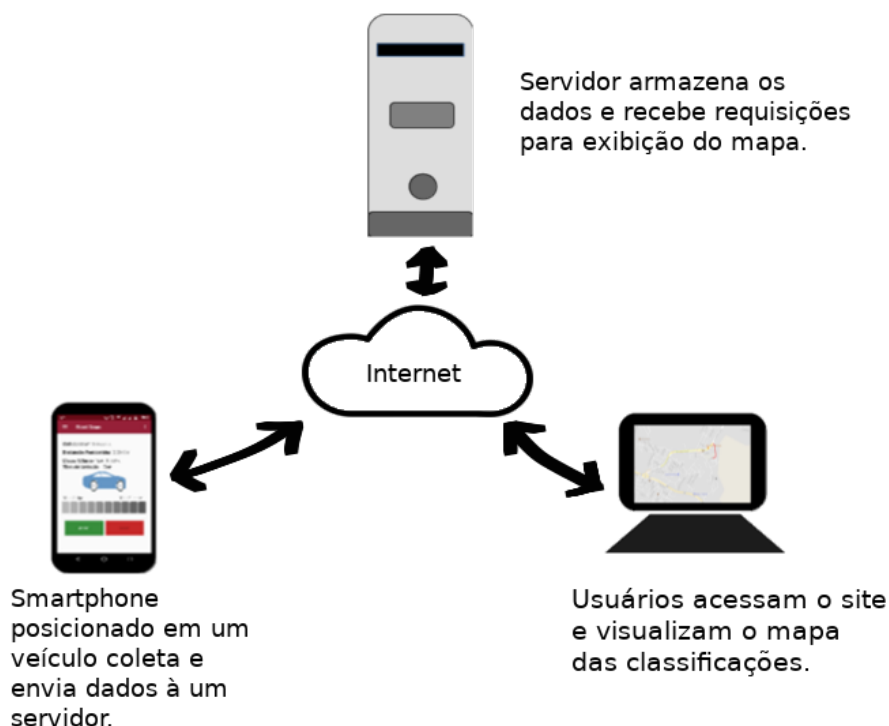
O sistema como um todo continuou se baseando na estratégia desenvolvida por ([LIMA et al., 2016](#)). Uma visão geral do mesmo pode ser observada na [Figura 7](#).

Abaixo serão explicadas de maneira aprofundada a responsabilidade e o funcionamento de cada um dos componentes desta arquitetura.

### 3.4.1 Smartphone

O aplicativo atual do RoadScan sofreu inúmeras modificações em relação a versão anterior desenvolvida por ([LIMA et al., 2016](#)). O aplicativo foi praticamente refeito,

Figura 7 – Arquitetura do sistema.



Fonte: Elaborado pelo autor.

aproveitando apenas a lógica da coleta dos dados do acelerômetro e a rotação dos eixos responsável por fazer com que sempre seja utilizado o eixo perpendicular a Terra como eixo Z do *smartphone*.

Nesta subseção será abordada o funcionamento do caso de sucesso principal referente ao uso do RoadScan que está representado também no diagrama da Figura 8.

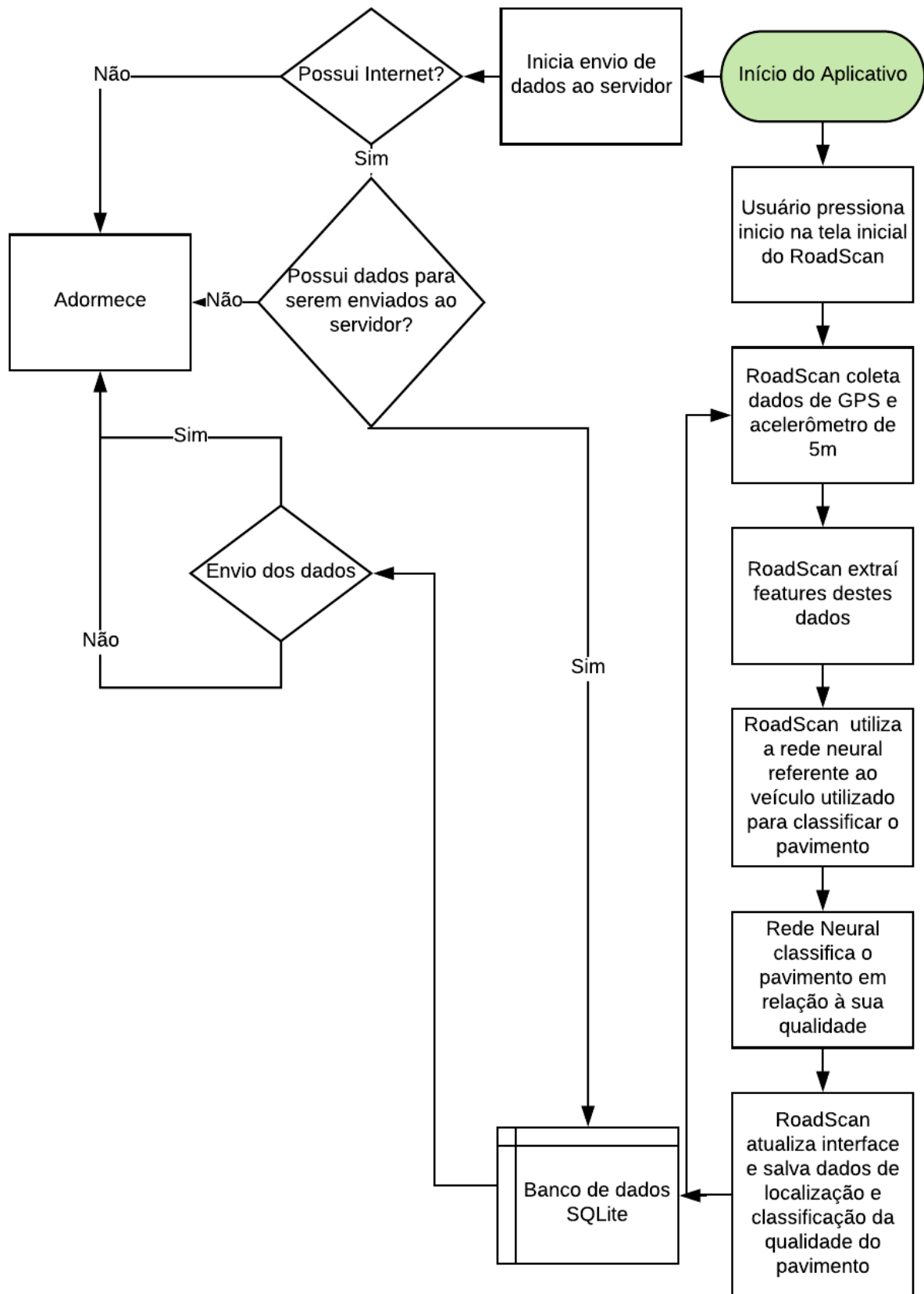
O usuário ao iniciar o aplicativo é solicitado a aceitar os termos de privacidade do sistema, visto que seus dados de localização, nome, *email* serão compartilhados com um servidor. Para efetuar o *login* o usuário poderá optar por utilizar sua conta do Google<sup>7</sup> ou ainda efetuar o *login* anônimo, preservando seu nome e *email*.

Efetuada o *login*, o usuário é encaminhado a tela inicial do aplicativo mostrada na Figura 9. Nesta tela o usuário possui acesso também a um menu, responsável por conceder acesso à outras funcionalidades do aplicativo como coleta de dados para treinamento de outra rede neural e visualização do mapa das classificações.

Na tela principal pode-se iniciar o serviço do RoadScan responsável por coletar os dados e realizar a classificação dos pavimentos. Quando o serviço é iniciado, o RoadScan verifica se o veículo já foi identificado previamente, caso não tenha sido, o sistema inicia uma

<sup>7</sup> Google- <<https://www.google.com.br>>

Figura 8 – Funcionamento do aplicativo no *smartphone*.



Fonte: Elaborado pelo autor.

Figura 9 – Tela principal do aplicativo RoadScan.



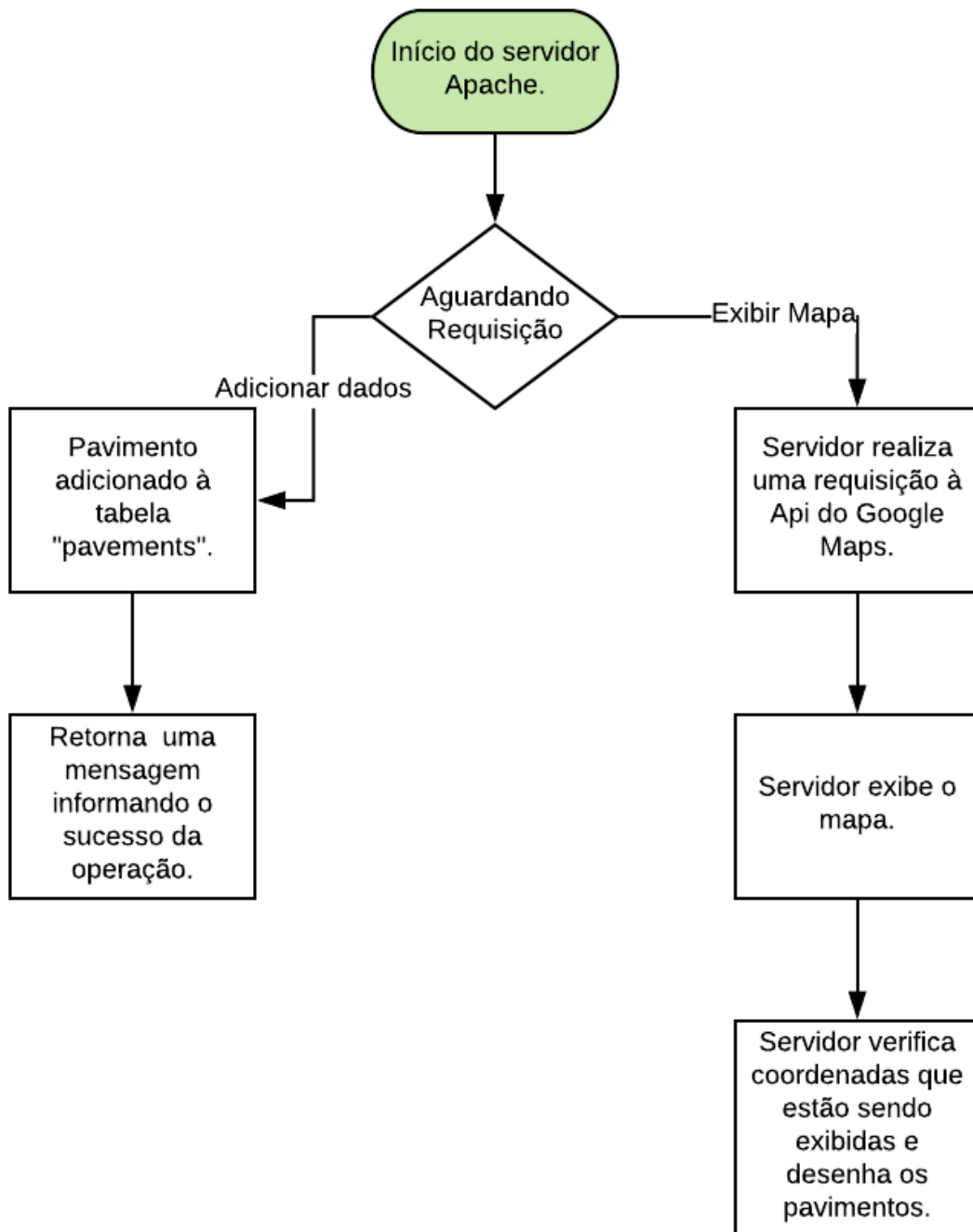
Fonte: Elaborado pelo autor.

tentativa de reconhecimento do veículo utilizando uma rede neural treinada exclusivamente para isto. Após 10 pavimentos de 5 metros ele gera uma classificação baseado na moda destas 10 predições. O usuário é então questionado quanto a assertividade da predição, onde caso esteja incorreta é solicitado que o mesmo informe o veículo e é contabilizado para trabalhos futuros que uma classificação incorreta foi realizada.

Selecionado então o veículo correto, o RoadScan começa a coletar os dados do acelerômetro e do [GPS](#), possibilitando com que se realize a classificação. Os pavimentos são separados de 5 em 5 metros. Após alcançada esta distância, 11 diferentes características são atualmente extraídas. A rede neural já treinada referente aquele veículo é acionada e calcula a qualidade do pavimento exibindo-a na tela do aplicativo.

Após este processo ser efetuado, os dados deste pavimento e sua localização também são adicionados à um banco de dados SQLite. Por fim estes dados são enviados ao servidor através de uma *thread* que é inicializada com a aplicação. Esta *thread* possui como função verificar se existem dados para serem enviados ao servidor e realizar o *upload* dos mesmos.

Figura 10 – Funcionamento do sistema no lado do servidor.



Fonte: Elaborado pelo autor.

### 3.4.2 Servidor

O RoadScan conta com um servidor responsável por armazenar os pavimentos juntamente com as classificações coletadas no aplicativo Android. Neste servidor existem basicamente duas formas de requisições: as requisições para se adicionar pavimentos ao banco de dados e as requisições para exibição do mapa.

### 3.4.2.1 Adição de pavimentos na base de dados do servidor

Como dito anteriormente existe uma *thread* na aplicação Android responsável por fazer o *upload* dos dados ao servidor. Para realizar este *upload*, lotes contendo 50 pavimentos são encapsulados para o formato JSON e enviados via requisição POST ao servidor. Quando estes dados chegam ao servidor eles são inseridos na tabela “*pavements*”, que armazena os pavimentos a serem exibidos no mapa.

### 3.4.2.2 Requisições do mapa

Para exibição do mapa com as classificações, o servidor utiliza a Google Maps API<sup>8</sup> para JavaScript. Durante a exibição dos dados o servidor verifica as coordenadas exibidas na tela e baseado nestas coordenadas consulta no banco de dados os pavimentos daquela área.

Estas consultas geram dados em locais extremamente próximos, que geralmente pertencem à diferentes usuários e são utilizados para o cálculo da média das classificações. Esta média é então exibida gerando uma informação que tende a ser mais precisa do que a informação gerada por apenas um veículo.

### 3.4.3 Visualização do Mapa por Usuários Finais

Para os usuários finais visualizarem o mapa com as classificações dois meios são disponibilizados: através do aplicativo RoadScan<sup>9</sup> e através do Site<sup>10</sup>.

<sup>8</sup> <<https://developers.google.com/maps/documentation/javascript/?hl=pt-br>>

<sup>9</sup> <[https://play.google.com/store/apps/details?id=com.samuel.roadscan&hl=pt\\_BR](https://play.google.com/store/apps/details?id=com.samuel.roadscan&hl=pt_BR)>

<sup>10</sup> <<http://200.239.153.214/roadie/roadScan.php>>

## 4 Resultados

Neste capítulo serão discutidos os resultados experimentais e práticos obtidos neste trabalho.

### 4.1 Resultados Experimentais

Os resultados experimentais aqui discutidos foram obtidos através da ferramenta Weka, utilizada como forma de validar a base de dados criada para este projeto e descrita na Seção 3.3. Serão apresentados também a matriz de confusão e outros dados referentes ao modelo final, representado pela rede neural treinada desenvolvida em Python.

Utilizou-se três algoritmos para validar a base de dados, são estes o SVM(CORTES; VAPNIK, 1995), a rede neural *multi-layer perceptron* (MLP) e o método de Naive Bayes(MCCALLUM; NIGAM et al., 1998). A validação cruzada foi utilizada para gerar as estatísticas e as matrizes de confusão. A base de dados foi dividida em 10 subconjuntos gerados aleatoriamente, e durante 10 vezes o algoritmo realizou o treinamento com 9 destes subconjuntos, efetuando os testes com o décimo subconjunto. Cada subconjunto serviu como conjunto de testes uma vez. Ao final os resultados foram condensados para gerar as estatísticas abordadas nessa seção.

As matrizes de confusão obtidas pelo Weka se encontram no Anexo B. Na linha das matrizes de confusão tem-se a verdadeira classificação dos dados, e as colunas representam a qualificação atribuída pelo modelo utilizado. Um modelo que represente exatamente a base de dados deveria conter valores apenas na diagonal principal desta matriz. Porém, isto nem sempre é desejado pois conforme afirmado na Seção 3.3, a base de dados gerada provavelmente conta com dados ruidosos devido a dificuldade em se encontrar trechos longínquos que possuam a mesma classificação. Assim, caso um modelo atinja uma matriz de confusão perfeita, significaria que o mesmo simplesmente se adequou ao ruído, o que não representa uma precisão do mundo real. Tendo em vista isto, os dados obtidos foram satisfatórios em praticamente todos os algoritmos.

Uma comparação melhor destes algoritmos pode ser observada na Tabela 1. Nela são apresentadas a precisão média de cada algoritmo, sua taxa média de Falsos Positivos (FP) que representa o quanto as amostras de uma classe diferente de X foram classificadas como X, taxa média de Verdadeiros Positivos (VP) que representa o quanto as amostras da classe X foram classificadas como X e a *f-measure* que é uma medida interessante para comparar os modelos gerados(POWERS, 2011).

Como dito anteriormente, uma matriz de confusão perfeita não é o desejado, porém,

ainda assim é interessante escolher um método em que se maximize a precisão do modelo pois, a base de dados apesar de conter ruídos, possui em sua maioria dados não ruidosos e que representam a classificação da qualidade dos pavimentos<sup>1</sup>.

Logo, baseando-se na comparação destes resultados onde a rede **MLP** teve uma melhor *f-measure*, menor índice de falsos positivos e maior índice de verdadeiros positivos, foi possível validar a base de dados e dar continuidade na pesquisa utilizando a rede neural **MLP** como classificador.

Tabela 1 – Comparação entre alguns algoritmos classificadores implementados no Weka utilizando a base de dados coletada neste trabalho.

<b>Classificador</b>	<b>Precisão Média</b>	<b>F-Measure Médio</b>	<b>VP Médio</b>	<b>FP Médio</b>
Rede Neural MLP	77.90%	0.778	0.779	0.082
SVM	76.23%	0.759	0.762	0.095
Naive Bayes	69.39%	0.683	0.694	0.115

Fonte: Elaborado pelo autor.

Os testes no Weka foram utilizados para auxiliar na escolha do classificador e na validação da base de dados. Após coletados os resultados implementou-se a rede neural **MLP** em Python e o modelo gerado alcançou 76% de precisão, onde toda a base de dados foi utilizada para se gerar o modelo. Na Tabela 2 pode-se observar que os maiores índices de “confusão” estão entre as classes mais próximas. A Tabela 3 apresenta as porcentagens de falsos positivos e negativos, e verdadeiros positivos e negativos de cada uma das classes. Além disto, ela mostra outras métricas interessantes como acurácia, revocação e a precisão. A acurácia representa no geral o quão frequente o modelo está correto, a precisão retrata quanto das amostras classificadas como corretas realmente estavam e a revocação reflete o quão frequente uma dada classe é classificada corretamente. Os resultados exibidos nesta tabela confirmam que o modelo obtido foi satisfatório, desta maneira foi possível utilizá-lo na aplicação final para classificar a qualidade dos pavimentos.

Tabela 2 – Matriz confusão obtida com o algoritmo rede neural MLP no modelo final.

	<b>Ótimo</b>	<b>Bom</b>	<b>Ruim</b>	<b>Péssimo</b>
<b>Ótimo</b>	268	95	5	0
<b>Bom</b>	83	292	8	0
<b>Ruim</b>	10	28	683	72
<b>Péssimo</b>	3	16	162	307

Fonte: Elaborado pelo autor.

<sup>1</sup> Baseado na classificação dada pelo autor.



Tabela 3 – Tabela de dados classificação do modelo final.

	Excelente		Bom		Ruim		Horrrível		Média	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Positivos	72,8%	27,1%	75,1%	24,9%	86,1%	13,9%	62,9%	37,1%	74,2%	25,8%
Negativos	5,7%	94,2%	8,0%	92,0%	13,5%	86,5%	6,5%	93,5%	8,5%	91,5%
Acurácia	0,905		0,889		0,863		0,866		0,881	
Precisão	0,724		0,677		0,796		0,734		0,733	
Revocação	0,728		0,750		0,861		0,629		0,742	
F-Score	0,726		0,712		0,827		0,677		0,736	

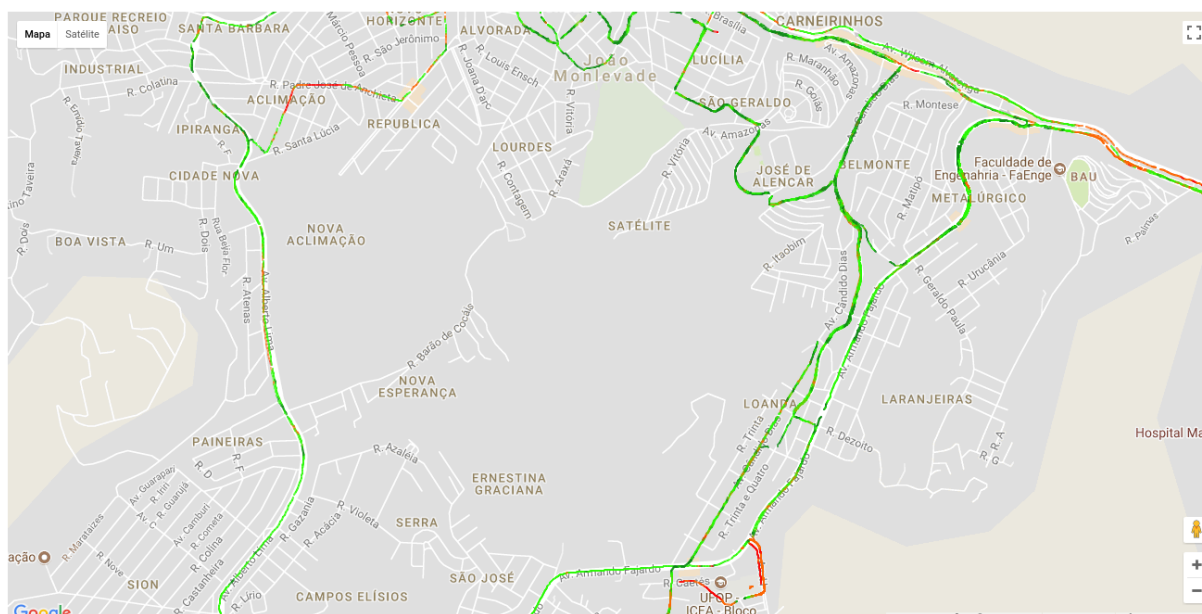
Fonte: Elaborado pelo autor.

## 4.2 Resultados Finais

A versão final do aplicativo foi lançada para um grupo pequeno de usuários que realizaram testes durante algumas semanas, isto possibilitou a detecção e correção de alguns erros da aplicação.

A Figura 11 apresenta um trecho do mapa de classificações gerado pelo tráfego de automóveis que utilizaram o RoadScan. Nota-se que as classificações obtidas refletem bem a qualidade das estradas e que apesar de algumas classificações falhas o resultado é satisfatório.

Figura 11 – Mapa classificações geradas pelo RoadScan.



Fonte: Mapa desenhado utilizando a Google Maps API com dados coletados e calculados pelo RoadScan.

## 5 Conclusão

A melhoria para o RoadScan desenvolvida neste trabalho visou aprimorar a precisão do trabalho anterior, mantendo as características interessantes como: apresentação da classificação ao usuário com um baixo tempo de resposta e o uso de técnicas de *crowdsourcing*. Além disto, este trabalho tornou possível que as características das classificações pudessem ser facilmente alteradas de acordo com as necessidades de um usuário final, ou seja, este agora pode participar do processo de desenvolvimento e dizer que tipo de pavimento ele considera péssimo, ruim, bom e ótimo, para que o sistema passe a classificar novos pavimentos de acordo com essas definições.

Os dados disponibilizados neste trabalho foram coletados na cidade de João Monlevade no estado de Minas Gerais e as classificações foram baseadas nas considerações do autor. Porém, como dito anteriormente, nada impede com que uma outra versão do aplicativo possa ser adaptada para as características e necessidades de outra localidade ou outro usuário caso as aqui definidas não sejam adequadas. Durante este trabalho tentou-se também realizar a diferenciação de veículos porém, os resultados não foram suficientemente satisfatórios.

Algumas dificuldades enfrentadas no trabalho anterior permanecem como desafios futuros, como por exemplo: freadas e acelerações bruscas interferindo na qualidade da classificação, além das lombadas como já citadas anteriormente. Inúmeras outras variáveis externas atrapalham as leituras do acelerômetro, e sua identificação, bem como uma maneira de inseri-las na classificação final deve ser estudado. Este trabalho apesar de não alcançar o patamar de solução perfeita, serve como um degrau para que este objetivo possa ser alcançado.

A utilização dos *smartphones* é algo que se tornou parte do cotidiano e cada vez mais temos dispositivos mais poderosos e com preços acessíveis. Isto possibilita a expansão do RoadScan, proporcionando ainda mais benefícios aos usuários do sistema, ajudando a solucionar um dos maiores desafios: motivar a utilização por parte dos colaboradores.

Para trabalhos futuros pode-se citar:

- Implementar técnicas de gamificação para beneficiar usuários que contribuam com o projeto.
- Identificar novas características que possam auxiliar na precisão do modelo.
- Expandir o sistema efetivamente para outros tipos veículos.
- Implementar atualização remota do modelo da rede neural presente no aplicativo.

## Referências

- BADUROWICZ, M.; CIEPLAK, T.; MONTUSIEWICZ, J. T. On-the-fly community-driven mobile accelerometer data analysis system for road quality assessment. *Applied Computer Science*, v. 12, n. 4, p. 18–27, 2016. Disponível em: <<http://acs.pollub.pl/pdf/v12n4/2.pdf>>. Citado na página 19.
- BISHOP, C. M. *PATTERN RECOGNITION AND MACHINE LEARNING*. [S.l.]: SPRINGER-VERLAG NEW YORK, 2006. Citado 2 vezes nas páginas 21 e 22.
- BRAGA, A. d. P.; CARVALHO, A. C. P. d. L. F.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicacoes*. [S.l.]: LTC Editora, 2007. Citado na página 22.
- CASALE, P.; PUJOL, O.; RADEVA, P. Human activity recognition from accelerometer data using a wearable device. *Pattern Recognition and Image Analysis Lecture Notes in Computer Science*, p. 289–296, 2011. Citado na página 31.
- CHUGH, G.; BANSAL, D.; SOFAT, S. Road Condition Detection Using Smartphone Sensors : A Survey. *International Journal of Electronic and Electrical Engineering*, v. 7, n. 6, p. 595–601, 2014. Disponível em: <[http://www.ripublication.com/irph/ijeee{\\\_}spl/ijeeev7n6{\\\_}](http://www.ripublication.com/irph/ijeee{\_}spl/ijeeev7n6{\_})>. Citado 2 vezes nas páginas 16 e 18.
- CNT. *Boletim Estatístico*. 2017. Disponível em: <<http://www.cnt.org.br/Boletim/boletim-estatistico-cnt>>. Citado 3 vezes nas páginas 8, 9 e 16.
- CORTES, C.; VAPNIK, V. Support-Vector Networks. v. 297, p. 273–297, 1995. Citado 3 vezes nas páginas 18, 21 e 38.
- EBRAHIMI, E.; MOLLAZADE, K.; AREFI, A. An expert system for classification of potato tubers using image processing and artificial neural networks. *International Journal of Food Engineering*, v. 8, n. 4, 2012. Citado na página 23.
- GARTNER. *Mobile OS market share 2017*. 2017. Disponível em: <<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>>. Citado na página 20.
- GEMAN, S.; BIENENSTOCK, E.; DOURSAT, R. Neural networks and the bias/variance dilemma. *Neural Computation*, v. 4, n. 1, p. 1–58, 1992. Disponível em: <<https://doi.org/10.1162/neco.1992.4.1.1>>. Citado na página 31.
- GEORGE, K. P.; RAJAGOPAL, A. s. Prediction models for pavement deterioration. *Pavement Asset Management*, p. 151–158, Jan 2015. Citado na página 16.
- KLAUBERT, E. C. *Highway effects on vehicle performance*. [S.l.]: Federal Highway Administration, 2001. Citado 3 vezes nas páginas 8, 9 e 16.
- LIMA, L. C. et al. Using crowdsourcing techniques and mobile devices for asphaltic pavement quality recognition. *2016 VI Brazilian Symposium on Computing Systems Engineering (SBESC)*, 2016. Citado 4 vezes nas páginas 8, 9, 18 e 32.

- MCCALLUM, A.; NIGAM, K. et al. A comparison of event models for naive bayes text classification. In: CITESEER. *AAAI-98 workshop on learning for text categorization*. [S.l.], 1998. v. 752, n. 1, p. 41–48. Citado na página 38.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 21.
- MEIRELLES, F. S. *28ª Pesquisa Anual do Uso de TI*. 2017. Disponível em: <<http://eaesp.fgv.br/ensinoeconhecimento/centros/cia/pesquisa>>. Citado na página 16.
- MINSKY, M. L.; PAPER, S. Perceptrons and pattern recognition. Jan 1967. Citado na página 22.
- MOHAMED, S. et al. Dynamic Mapping of Road Conditions Using Smartphone Sensors and Machine Learning Techniques. p. 1–5, 2016. Citado na página 19.
- MOHAN, P.; PADMANABHAN, V. N.; RAMJEE, R. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys 08*, 2008. Citado na página 16.
- NOVAC, O. C. et al. Comparative study of Google Android, Apple iOS and Microsoft Windows Phone mobile operating systems. *2017 14th International Conference on Engineering of Modern Electric Systems, EMES 2017*, p. 154–159, 2017. Citado na página 21.
- POWERS, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. Bioinfo Publications, 2011. Citado na página 38.
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, 1958. Citado na página 19.
- ROSSUM, G. v. *The History of Python*. 2009. Acessado em: 2018-02-01. Disponível em: <<http://python-history.blogspot.com.br/2009/01/introduction-and-overview.html>>. Citado na página 25.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, 1986. Citado na página 22.
- SAYERS, M. W. The international road roughness experiment: establishing correlation and a calibration standard for measurements. University of Michigan, Ann Arbor, Transportation Research Institute, 1986. Citado na página 16.
- SERAJ, F. et al. Roads: A road pavement monitoring system for anomaly detection using smart phones. *Lecture Notes in Computer Science Big Data Analytics in the Social and Ubiquitous Context*, p. 128–146, 2016. Citado 2 vezes nas páginas 18 e 31.
- STALLINGS, W. *Operating Systems: Internals and Design Principles*. 6th. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008. ISBN 0136006329, 9780136006329. Citado na página 20.
- TENSORFLOW. 2018. Acessado em: 2018-02-01. Disponível em: <<https://www.tensorflow.org/>>. Citado na página 25.

WANG, R. Y.; CHUANG, Y. T.; YI, C. W. A crowdsourcing-based road anomaly classification system. *18th Asia-Pacific Network Operations and Management Symposium, APNOMS 2016: Management of Softwarized Infrastructure - Proceedings*, 2016. Citado na página 19.

WEKA. *Weka 3: Data Mining Software in Java*. 2018. Acessado em: 2018-02-01. Disponível em: <<https://www.cs.waikato.ac.nz/ml/weka/>>. Citado na página 25.

WU, X. *Knowledge acquisition from databases*. [S.l.]: Ablex, 1995. Citado na página 23.

ZACCONE, G. *Deep learning with TensorFlow*. [S.l.]: Packt Publishing, 2017. Citado na página 26.

ZHU, X.; WU, X. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, v. 22, n. 3, p. 177–210, 2004. Citado na página 24.

# Anexos

# ANEXO A – Features Utilizadas

Abaixo tem-se uma breve descrição das *features* utilizadas no RoadScan.

## A.1 Média

Métrica utilizada para apontar a tendência do centro de um conjunto de dados.

$$\mu_s = \frac{1}{N} \sum_{i=1}^N s_i \quad (\text{A.1})$$

## A.2 Desvio Padrão

Valor não negativo que representa o quão próximos da média os elementos de um conjunto estão.

$$\sigma_s = \sqrt{\frac{1}{N} \sum_{i=1}^N (s_i - \mu_s)^2} \quad (\text{A.2})$$

## A.3 Coeficientes de variação

Medida utilizada que representa o desvio padrão expresso como porcentagem da média.

$$c_v = \frac{\sigma_s}{\mu_s} \quad (\text{A.3})$$

## A.4 Diferença pico a pico

Diferença entre o maior e o menor valor de um conjunto de dados.

$$\delta_p = \max s_i - \min s_i \quad (\text{A.4})$$

## A.5 Obliquidade(Skewness)

Medida de assimetria de uma distribuição de frequência.

$$\text{Obliquidade} = \frac{\frac{1}{N} \sum_{i=1}^N (s_i - \mu_s)^3}{\left(\frac{1}{N} \sum_{i=1}^N (s_i - \mu_s)^2\right)^{\frac{3}{2}}} \quad (\text{A.5})$$

## A.6 Curtose(Kurtosis)

a medida de forma que caracteriza o achatamento da curva da função de distribuição de probabilidade

$$Curtose = \frac{\frac{1}{N} \sum_{i=1}^N (s_i - \mu_s)^4}{\left(\frac{1}{N} \sum_{i=1}^N (s_i - \mu_s)^2\right)^2} - 3 \quad (\text{A.6})$$

## A.7 Energia

Medida de um sinal representada pela equação abaixo.

$$\sum_{i=1}^N |s_i|^2 \quad (\text{A.7})$$

## A.8 Potência

Medida de um sinal representada pela equação abaixo.

$$\frac{\sum_{i=1}^N |s_i|^2}{N} \quad (\text{A.8})$$

## A.9 Raiz do valor quadrático médio

Medida da magnitude de um conjunto de dados.

$$x_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (\text{A.9})$$

## A.10 Velocidade

Média da velocidade obtida em cada pavimento.



## ANEXO B – Matriz de confusão dos algoritmos testados no *Weka*

Neste anexo serão apresentadas as matrizes de confusão obtidas por três algoritmos distintos de *machine learning* implementados pelo *Weka*. Os algoritmos foram utilizados para testar a base de dados criada para representar os quatro tipos de pavimentos.

Tabela 4 – Tabela confusão obtida com o algoritmo *SVM* implementado pelo *Weka*.

	<b>Ótimo</b>	<b>Bom</b>	<b>Ruim</b>	<b>Péssimo</b>
<b>Ótimo</b>	287	78	3	0
<b>Bom</b>	106	253	24	0
<b>Ruim</b>	18	5	696	74
<b>Péssimo</b>	4	11	160	313

Tabela 5 – Matriz confusão obtida com o algoritmo rede neural *MLP* implementado pelo *Weka*.

	<b>Ótimo</b>	<b>Bom</b>	<b>Ruim</b>	<b>Péssimo</b>
<b>Ótimo</b>	284	79	4	1
<b>Bom</b>	94	269	12	8
<b>Ruim</b>	6	23	683	81
<b>Péssimo</b>	4	15	122	347

Tabela 6 – Matriz confusão obtida com o algoritmo de Naive Bayes implementado pelo *Weka*.

	<b>Ótimo</b>	<b>Bom</b>	<b>Ruim</b>	<b>Péssimo</b>
<b>Ótimo</b>	324	38	6	0
<b>Bom</b>	220	145	17	1
<b>Ruim</b>	6	19	648	120
<b>Péssimo</b>	4	12	179	293