

**Universidade do Minho**  
Escola de Ciências

Diana Sofia Araújo Carrilho

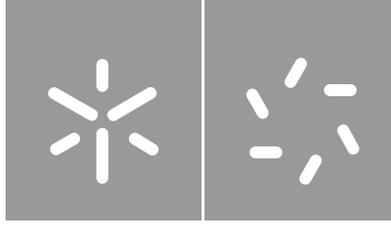
## **Sinais de Trânsito Inteligentes**

Diana Carrilho **Sinais de Trânsito Inteligentes**

UMinho | 2020

dezembro de 2020





**Universidade do Minho**

Escola de Ciências

Diana Sofia Araújo Carrilho

## **Sinais de Trânsito Inteligentes**

Dissertação de Mestrado  
em Matemática e Computação

Trabalho efetuado sob a orientação da  
**Professora Doutora Ana Jacinta Pereira da  
Costa Soares**  
e da  
**Professora Doutora Maria Fernanda Pires da  
Costa**

## **Direitos de Autor e Condições de Utilização de Trabalho por Terceiros**

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos. Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada. Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.



### **Atribuição-NãoComercial-Compartilhalgual CC BY-NC-SA**

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

## **Agradecimentos**

O presente trabalho é o resultado de dois longos e intensos anos que me ajudaram a enriquecer quer pessoal quer profissionalmente. Durante este tempo tive a oportunidade de conhecer novas áreas, novos caminhos e novas pessoas que ficarão no meu coração. Por isso, quero agradecer às mais importantes que sempre estiveram prontas a ajudar-me e a partilharem comigo os seus conhecimentos.

Um agradecimento muito especial aos meus orientadores, Professora Doutora Ana Jacinta Soares, Professora Doutora Maria Fernanda Costa e ao Professor Doutor Luís Ferrás que contribuíram para que este trabalho ficasse com rigor, perante muita persistência, disponibilidade e sentido crítico. Para mim, foi um privilégio tê-los como orientadores. Nunca vou esquecer a vossa profissionalismo e humildade em estarem sempre prontos a ajudar.

Um agradecimento à *Accenture*, por me ter proporcionado a oportunidade de desenvolvimento deste projecto, em particular, ao Rui Pires pela sua imaginação e conhecimentos que me levaram a construir este trabalho.

Um grande agradecimento aos meus colegas à Cecília e ao Luís pela amizade e por todos os bons momentos que passamos. Ficarei com dois grandes e bons amigos para a vida.

Um muito obrigado ao meu grande amor, Jorge, pela paciência e apoio. Sem ele, seria muito mais difícil ultrapassar este desafio.

Finalmente, e não menos importante, quero terminar com um agradecimento à minha família, nomeadamente aos meus tios, Conceição e Manuel que me acolheram há 12 anos para que pudesse ser a pessoa que sou hoje. Ao meu pai e aos meus avós, muito obrigada pelos bons princípios que me transmitiram.

*À minha mãe...*



## **Declaração de Integridade**

Declaro ter actuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração. Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.



## Resumo

### Sinais de Trânsito Inteligentes

Com o aumento do número de veículos nas cidades, os semáforos convencionais de tempos fixos não conseguem escoar todos os veículos que chegam aos cruzamentos, levando a que os automobilistas fiquem retidos nas filas por longos períodos de tempo. Como consequência, são originados inúmeros problemas de congestionamento de tráfego que têm um impacto negativo na vida dos condutores, nas cidades, no meio ambiente do planeta e na saúde mental das populações.

Nesta dissertação, numa primeira parte, é desenvolvido e estudado um modelo que simula um cruzamento contendo semáforos com tempos fixos, designado por simulador convencional. Posteriormente, tendo por base o simulador convencional, foi desenvolvido um simulador manual que permite gerar e guardar dados de fluxo de tráfego que serão usados como conjuntos de dados de treino em redes neuronais.

Este simulador manual consiste numa alteração do simulador convencional, permitindo a um utilizador activar a luz verde, através do teclado do computador, consoante o número de veículos que se encontram nas filas de espera do cruzamento de maneira a escoar o fluxo de tráfego.

Numa segunda parte, tendo novamente por base o simulador convencional, foi desenvolvido um simulador inteligente. O mecanismo inteligente foi implementado com abordagem de *Machine Learning*, mais concretamente com abordagem de *Supervised Learning*, usando redes neuronais artificiais. O simulador inteligente foi usado para treinar e para testar as redes neuronais propostas nesta dissertação. Após o treino de uma rede neuronal o simulador começará a tomar decisões de qual o semáforo a abrir, de modo a tentar minimizar os tempos de espera bem como a tentar maximizar a fluidez do trânsito. Com estas técnicas, os semáforos serão activados consoante o número de veículos e o tempo médio de espera das filas, havendo assim uma gestão inteligente dos tempos de verde e de vermelho.

**Palavras-chave:** Intersecção Inteligente, *Machine Learning*, *Supervised Learning*, Semáforo.



## **Abstract**

### **Sinais de Trânsito Inteligentes**

With the increase in the number of vehicles, conventional fixed-time traffic lights are not able to drain all vehicles arriving at intersections, causing drivers to be stuck in traffic for long periods of time. As a consequence, numerous traffic congestion problems originate, that have a negative impact on the lives of drivers, cities, on the planet's environment and on the mental health of the populations.

In this thesis, a model for the implementation of an intersection containing traffic lights with fixed times is presented. This model is called the conventional simulator. Using this simulator, a manual simulator was subsequently developed, allowing to generate and save traffic flow data, that will be used as training data sets in neural networks. This manual simulator consists of a modification to the conventional simulator, allowing the user to activate the green light, via the computer keyboard, depending on the number of vehicles that are in the waiting queues at the intersection, in order to drain the traffic flow.

Based again on the conventional simulator, an intelligent simulator was developed. The intelligent mechanism was implemented with a Machine Learning approach, more specifically with a Supervised Learning approach, using artificial neural networks. The intelligent simulator was used to train and test the neural networks proposed in this dissertation. After training a neural network, the simulator will start making decisions about which traffic light to open, in order to minimize waiting times as well as to maximize the traffic fluidity. With these techniques, traffic lights will be activated depending on the number of vehicles and the average waiting time of the queues, thus making use of an intelligent management of green and red times.

**Keywords:** Intelligent Intersection, Machine Learning, Supervised Learning, Traffic Light.



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Visão Geral de Sistemas de Controlo de Tráfego</b>	<b>4</b>
2.1	Métodos para Detecção de Tráfego	4
2.1.1	Métodos de <i>Floating Car Data</i>	4
2.1.2	Métodos que usam sensores	5
2.1.3	Aplicações	6
2.2	Métodos para controlo de fluxo de tráfego	8
2.2.1	Sistema de Controlo de Tráfego SCOOT	8
2.2.2	Sistema de Controlo de Tráfego GERTRUDE	9
2.2.3	Sistema de Controlo de Tráfego UTMS	10
2.3	Parâmetros, Variáveis e Modelação	10
2.3.1	Modelação Microscópica	11
2.3.2	Modelação Macroscópica	12
2.3.3	Modelação Mesoscópica	12
2.3.4	Parâmetros e Variáveis Fundamentais	13
<b>3</b>	<b>Simulador Convencional</b>	<b>16</b>
3.1	Descrição do Simulador Convencional	16
3.2	Variáveis do fluxo de tráfego	17
3.2.1	Tempo de Reacção ( $T_{reac}$ )	17
3.2.2	Velocidade Média ( $V_{med}$ )	17
3.2.3	Distância <i>Bumper-to-Bumper</i> ( $D_{btb}$ )	17
3.2.4	Comprimento total de um veículo ( $C_{tnei}$ )	18
3.2.5	Distância do veículo ao semáforo ( $D_{neisem}$ )	18
3.2.6	Tempos estimados ( $T_{verini}$ , $T_{fv}$ , $T_{tdeco}$ )	18
3.3	Resultados com o simulador	22
<b>4</b>	<b>Simulador Inteligente</b>	<b>23</b>
4.1	Descrição do simulador inteligente	24
4.2	Conjuntos de dados de treino	25
4.3	Redes Neurais Artificiais usadas	27
4.4	Treino das Redes Neurais	28
4.5	Métricas de desempenho	29
4.6	Validação dos Modelos	32
4.6.1	Modelos- $T_{med}$ , $T_{med-50\%vec1}$ , $T_{med-70\%vec1}$ , $T_{med-70\%mf}$	33
4.6.2	Validação dos Modelos- $T_{med}$ , $T_{med-50\%vec1}$ , $T_{med-70\%vec1}$ , $T_{med-70\%mf}$ treinados com dados de reforço	39
4.6.3	Comparação dos Modelos treinados sem e com dados de reforço	43
4.7	Uso do desvio-padrão do tempo médio de espera como <i>input</i>	44

<b>5</b>	<b>Simulador Convencional versus Simulador Inteligente</b>	<b>47</b>
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>50</b>
6.1	Conclusões . . . . .	50
6.2	Trabalho Futuro . . . . .	52

## Lista de Figuras

1	Esquema de um cruzamento com quatro semáforos [13]. . . . .	2
2	Representação do método da Triangulação (figura adaptada de [34]). . . . .	4
3	Representação do método baseado em GPS (figura adaptada de [12]). . . . .	5
4	Arquitetura de sensores indutivos (figura adaptada de [14]). . . . .	6
5	Arquitetura do sistema <i>eCall</i> (figura adaptada de [11]). . . . .	7
6	Arquitetura do sistema SCOOT (figura adaptada de [28]). . . . .	9
7	Arquitetura do sistema GERTRUDE (figura adaptada de [9]). . . . .	10
8	Modelo car-Following (figura adaptada de [4]). $c$ é o comprimento do veículo, $x_n$ a sua posição e $v_n$ a sua velocidade. . . . .	12
9	Níveis dos três tipos de modelação (figura adaptada de [23]). . . . .	13
10	Esquema de um ciclo com 4 fases. . . . .	13
11	Gráfico do Fluxo de Saturação (figura adaptada de [36]). . . . .	14
12	Faixas A, B, C e D. . . . .	16
13	Distância <i>bumper-to-bumper</i> . . . . .	17
14	Distâncias <i>bumper-to-bumper</i> consideradas (figura adaptada de [1]). . . . .	18
15	Esquema exemplificativo da aplicação da função $f$ definida em (10). . . . .	20
16	Interface gráfica do simulador. . . . .	21
17	Diagrama das técnicas de Inteligência Artificial (figura adaptada de [31]). . . . .	23
18	Polícia Sinaleiro na cidade de Braga [10]. . . . .	25
19	Parte do <i>Dataset</i> para treino da Rede Neuronal. . . . .	26
20	Desempenho da fase de treino da rede. . . . .	29
21	Modo de funcionamento do Perito e da rede neuronal, na optimização do fluxo de tráfego num cruzamento. . . . .	33
22	Menu para a escolha do tipo de Simulação Inteligente. . . . .	33
23	Variação da <i>Accuracy</i> perante os vários pesos com dados gerados de forma manual. . . . .	38
24	Esquema cronológico da construção de cada conjunto de treino. . . . .	39
25	Variação da <i>Accuracy</i> e <i>Overall Accuracy</i> perante os vários pesos com dados de reforço. . . . .	43
26	Parte do <i>Dataset</i> para treino do Modelo- $T_{med-\sigma}$ . . . . .	45
27	Evolução do trabalho desenvolvido nesta dissertação, desde a implementação de raiz do simulador convencional, até ao desenvolvimento do simulador inteligente. . . . .	47
28	Tempos médios de espera obtidos com os Modelos- $T_{med-50\%vec1}$ , $T_{med-\sigma}$ e o simulador convencional, para as quatro faixas do cruzamento. . . . .	48
29	Número médio de paragens obtido com os Modelos- $T_{med-50\%vec1}$ , $T_{med-\sigma}$ e o simulador convencional, para as quatro faixas do cruzamento. . . . .	49
30	Exemplo de uma intersecção da cidade de Braga [8]. . . . .	53



## Lista de Tabelas

1	Variáveis e expressões usadas na fórmula	19
2	Resultados da simulação com tempos de verde fixos	22
3	Matriz de confusão binária	30
4	Matriz de confusão multiclasse	30
5	Exemplo de uma matriz de confusão multi-classe de uma rede neuronal	31
6	Matriz de confusão para Modelo- $T_{med}$	34
7	Tabela das métricas para Modelo- $T_{med}$	34
8	Matriz de confusão para o Modelo- $T_{med-50\%vec1}$	35
9	Tabela das métricas para o Modelo- $T_{med-50\%vec1}$	35
10	Matriz de confusão para o Modelo- $T_{med-70\%vec1}$	36
11	Tabela das métricas para o Modelo- $T_{med-70\%vec1}$	36
12	Matriz de confusão para o Modelo- $T_{med-70\%mf}$	37
13	Tabela das métricas para o Modelo- $T_{med-70\%mf}$	37
14	Matriz de confusão para o Modelo- $T_{med}$	39
15	Tabela das métricas para o Modelo- $T_{med}$	40
16	Matriz de confusão para o Modelo- $T_{med-50\%vec1}$	40
17	Tabela das métricas para o Modelo- $T_{med-50\%vec1}$	41
18	Matriz de confusão para o Modelo- $T_{med-70\%vec1}$	41
19	Tabela das métricas para o Modelo- $T_{med-70\%vec1}$	42
20	Matriz de confusão para o Modelo- $T_{med-70\%mf}$	42
21	Tabela das métricas para o Modelo- $T_{med-70\%mf}$	43
22	Resultados da simulação com os modelos- $T_{med}$ , $T_{med-50\%vec1}$ , $T_{med-70\%vec1}$ , $T_{med-70\%mf}$ sem dados de reforço	44
23	Resultados da simulação com os modelos- $T_{med}$ , $T_{med-50\%vec1}$ , $T_{med-70\%vec1}$ , $T_{med-70\%mf}$ com dados de reforço	44
24	Resultados da simulação com o modelo- $T_{med-o}$	45



# 1 Introdução

O primeiro semáforo a ser colocado numa rua teve um tempo de vida muito curto devido a uma explosão. Era operado manualmente e funcionava a gás, tendo sido instalado em Londres, em Dezembro de 1868. O semáforo, tal como o conhecemos actualmente, foi implementado pela primeira vez em 1914 (há cerca de 106 anos) em Cleveland (USA) [20].

É de notar que a palavra *semáforo* deriva da composição dos termos gregos *sêma*, atos (sinal) e *phorós*, ós, ón (que conduz).

Nos semáforos actuais podemos encontrar três cores que estão distribuídas da seguinte forma: a luz vermelha está acima ou à esquerda, a luz verde está em baixo ou à direita, e a luz amarela fica no centro. A razão das três cores universalmente aceites é simples. O vermelho, na natureza, representa uma cor de aviso, alarme ou perigo. Portanto, quando uma sinalização se encontra com a luz vermelha os condutores são obrigados a parar. O maior contraste com o vermelho é a sua cor oposta, o verde, que transmite aos condutores a informação de que podem atravessar a intersecção ou o cruzamento, caso as condições da via assim o permitam. O amarelo foi uma cor que se incorporou mais tarde nos semáforos, e permite transmitir mais informação aos condutores ou peões. Esta cor surge na transição entre a luz verde para a luz vermelha, avisando os automobilistas para não atravessarem a zona que o semáforo está a delimitar, a menos que a paragem não seja possível de forma segura.

Tal como o seu nome diz, o semáforo é feito para *conduzir* o trânsito, para o guiar, e permitir colocar alguma ordem quando existem intersecções.

Uma possível alternativa ao semáforo é a rotunda, que também permite a distribuição do trânsito em cruzamentos. As rotundas são espaços de circulação rodoviária, com forma geralmente circular, com várias vias de tráfego, onde o trânsito se faz em sentido giratório. Existem vários tipos de rotundas, mas todas elas têm como objectivo principal aumentar a fluidez de trânsito e reduzir o tempo de acesso e de passagem para uma melhor qualidade da circulação.

O ideal seria dispormos de um conjunto variado de vias que permitissem ao condutor escolher a que pretende seguir quando deparado com um cruzamento, sem ter que parar ou abrandar. Na realidade, em alguns casos tal é possível. Porém, atendendo à estrutura e arquitetura dos centros urbanos, e ao custo elevado da construção de um conjunto de vias, o semáforo é, nos dias de hoje, considerado a melhor opção.

É de salientar que, o congestionamento do tráfego é um problema que a todos diz respeito de uma forma directa ou indirecta, isto é, podemos estar a fazer parte do congestionamento ou apenas a passear numa eco-pista de uma cidade e a respirar as emissões que os veículos emitem. Tal situação deve-se ao facto dos congestionamentos fazerem com que haja uma maior emissão de gases poluentes para a atmosfera. Este é um problema cada vez mais frequente, tendo vindo a aumentar com o decorrer do tempo.

O bom funcionamento das artérias de uma cidade é fundamental para o desenvolvimento da mesma, fazendo com que haja um bom crescimento económico para o emprego e estabilidade dos seus habitantes.

Com o avanço da tecnologia, e estando cada vez mais presente o tema das *smart cities*, é possível tornar o meio urbano mais sustentável, criando soluções inteligentes que ajudem à mobilidade das vias, tornando-as mais fluidas. Assim, se ao tópico das *smart cities* se associar técnicas de inteligência artificial conseguiremos obter uma solução mais adaptada a cada arquitectura urbana, conseguindo criar relações entre os componentes que interferem na artéria onde se pretende implementar tal solução.

Esta dissertação visa criar uma solução inteligente usando aprendizagem automática supervisionada, nomeadamente, redes neuronais artificiais, de forma a diminuir o tempo médio de espera dos automobilistas tornando o tráfego mais fluido num cruzamento com quatro semáforos e consequentemente evitando congestionamentos que levam a

problemas que afectam todo o planeta. O objectivo deste trabalho consiste em resolver o seguinte problema, representado esquematicamente na figura 1. O que se pretende é minimizar os tempos médios de espera dos veículos que se encontram nas quatro faixas representadas por  $F_i$  onde  $i = 1, \dots, 4$ .

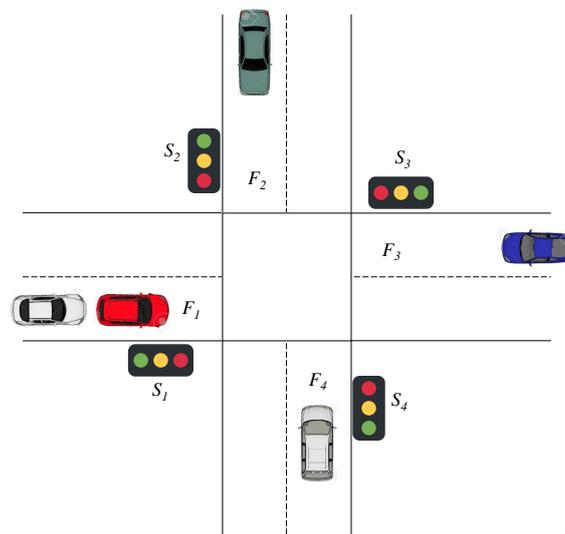


Figura 1: Esquema de um cruzamento com quatro semáforos [13].

Algumas perguntas que agora se colocam são: será que o tempo de verde deve ser igual ao tempo de espera no vermelho? Será que todos os semáforos de uma cidade devem ter o mesmo tempo de espera? Será possível um semáforo adaptar os tempos de cada cor, em tempo real, com base no número de veículos que aguardam na faixa respectiva?

A resposta a algumas destas perguntas é óbvia, e provavelmente já todos pensámos numa possível solução. A dificuldade está em saber qual a melhor combinação de intervalos de tempo para cada uma das cores do semáforo, como esta combinação varia ao longo do dia e como depende do local onde o semáforo está implementado. Podemos então depreender que este problema é mais complexo do que parece ser no primeiro instante. De facto, este tema tem interessado investigadores de várias áreas e tem merecido um estudo aprofundado. Será também o tema central do estudo desenvolvido nesta dissertação.

Para o desenvolvimento deste estudo foram criados três tipos de simuladores de tráfego. Para cada simulador, inicialmente, os dados de entrada são o número de veículos por faixa, que será o mesmo, para o simulador convencional, simulador manual (tempos de verde fixos) e simulador inteligente (tempos de verde variáveis conforme o número de veículos e os seus tempos médios de espera). Todo o processo aleatório de geração de parâmetros utilizados nas simulações será o mesmo para os três tipos de simuladores, de modo a poder comparar os resultados obtidos com cada um dos simuladores. Numa primeira fase foi desenvolvido um simulador convencional. A seguir, utilizando parte do simulador convencional, foi desenvolvido um simulador manual que permite gerar os dados de fluxo de tráfego de forma manual e guardar esses dados para criar os conjuntos de dados de treino necessários para treinar as redes neuronais. Este simulador manual consiste numa modificação do simulador convencional, permitindo, a um utilizador, activar a luz verde, através do teclado do computador, consoante o número de veículos que se encontram nas filas de espera do cruzamento de forma a fazer o escoamento. Numa segunda fase, a partir também do simulador

convencional, foi desenvolvido um simulador inteligente. O simulador inteligente desenvolvido foi usado tanto para treinar como para testar as redes neuronais propostas nesta dissertação para implementar este simulador inteligente. Após o treino de uma rede neuronal o simulador começará a tomar decisões de qual o semáforo a abrir, de modo a tentar minimizar os tempos de espera bem como a maximizar a fluidez do trânsito.

O mecanismo inteligente desenvolvido tem como principal abordagem a aprendizagem automática supervisionada, nomeadamente, o uso de redes neuronais artificiais.

## 2 Visão Geral de Sistemas de Controlo de Tráfego

Neste capítulo, apresentam-se algumas metodologias existentes na literatura relacionadas com o controlo do fluxo de tráfego.

### 2.1 Métodos para Detecção de Tráfego

Para recolher informações sobre o volume de tráfego é, em geral, necessário existir uma infra-estrutura (*hardware* e *software*) preparada para recolher os dados adquiridos pelo sistema. Nesta secção apresentam-se alguns dos mecanismos e métodos mais usados para este fim.

#### 2.1.1 Métodos de Floating Car Data

Os métodos de *Floating Car Data* são uma importante fonte de recolha de dados de tráfego, devido ao seu reduzido custo, apesar dos problemas de credibilidade. Em geral, para a utilização deste tipo de procedimento, todos os veículos que se encontram na via devem estar equipados com sensores, que permitem o envio de informação para as centrais de controlo de tráfego. Na ausência de qualquer outra fonte de dados, estes métodos são uma excelente forma de visualizar as condições do estado do tráfego. Apresentam-se a seguir dois métodos do *Floating Car Data* mais usados [3].

1. Método da triangulação: Os métodos de triangulação permitem, caso existam equipamentos electrónicos no interior dos automóveis, por exemplo *smartphones*, que haja transmissão de informação. À medida que o carro se movimentar, o sinal do telemóvel que está no interior da viatura também fará o mesmo, enviando a informação aos receptores, que utilizam o método geométrico da triangulação, que calcula a sua distância a três satélites e deste modo consegue determinar a localização do emissor, como podemos ver na figura 2. Uma vantagem deste método é o facto de não ser necessária nenhuma infra-estrutura na via, uma vez se recorre à rede móvel. Porém, este método não é tão vantajoso quando nos referimos a áreas onde a mesma rede móvel serve estradas de várias direcções diferentes.



Figura 2: Representação do método da Triangulação (figura adaptada de [34]).

2. Método baseado em GPS: Para usar este método é necessário que cada viatura esteja equipada com GPS (do inglês *Global Positioning System*). O GPS está instalado no interior dos veículos, transmitindo os dados para os servidores através da rede de comunicações GSM/GPRS (do inglês *Global System for Mobile/General Packet Radio Service*), o que permite determinar a localização e seguir a trajetória desse veículo em tempo real. A posição do veículo detectada através do GPS é usada para determinar a sua velocidade e a estrada onde este se encontra. Este sistema está representado esquematicamente na figura 3;

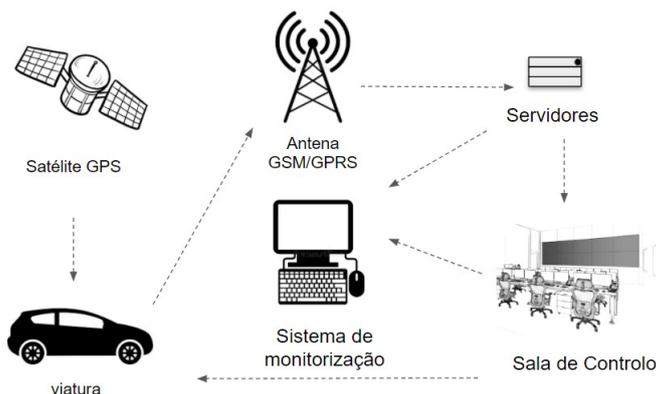


Figura 3: Representação do método baseado em GPS (figura adaptada de [12]).

Em geral, os sistemas de controlo de tráfego conhecem as velocidades médias praticadas em cada uma das estradas em condições de não congestionamento, e consequentemente, esperará que uma determinada velocidade média seja praticada pelos veículos que circulam nessas vias [25].

As vantagens dos sistemas *Floating Car Data* devem-se ao facto de serem de reduzido custo dado que recorrem a sensores e/ou câmaras, funcionam em zonas mais abrangentes, a sua configuração é simples, não necessitam de muita manutenção e funcionam em condições meteorológicas adversas.

Também existem outros métodos que permitem a identificação de viaturas através da sua matrícula. No entanto, estes métodos podem ser desvantajosos uma vez que é necessário a existência de infra-estruturas que permitam detectar a matrícula. O tempo de viagem e a velocidade do veículo poderão ser calculados comparando o tempo em que os pares de sensores (por exemplo, câmaras de vídeo em pórticos) detectam esse veículo. Estes métodos são apresentados na secção seguinte.

### 2.1.2 Métodos que usam sensores

Nesta subsecção são apresentados métodos que usam sensores para a recolha de informação sobre o fluxo de tráfego. Estes métodos podem ser divididos em **métodos intrusivos** e **métodos não-intrusivos**.

Os métodos intrusivos foram desenvolvidos para contabilizar o número de veículos, sendo os mais conhecidos designados por *loops* indutivos. Estes métodos necessitam de dispositivos para armazenar os dados e a sua implementação implica a instalação de sensores no pavimento da respectiva via.

Os métodos não intrusivos permitem monitorizar o tráfego usando câmaras de vídeo, sendo este tipo de métodos o mais usado. Estes métodos são mais vantajosos que os métodos intrusivos, uma vez que não necessitam de sensores que impliquem destruir e interromper a via para sua instalação.

Porém é de referir que as câmaras de vídeo estão mais sujeitas a actos de vandalismo. Mais ainda, a passagem de peões, animais ou outros objectos, pode levar à obtenção de dados errados.

Apresentam-se de seguida alguns destes métodos.

1. Detecção de veículos por *loop* indutivo: Este tipo de método intrusivo recorre à instalação de sensores/placas de metal debaixo do asfalto, com a finalidade de detectar veículos quando circulam pelo campo magnético correspondente, permitindo assim contarem os carros num dado intervalo de tempo. A principal desvantagem deste tipo de método deve-se ao facto da instalação dos sensores ser de elevado custo e morosa.

Os sensores mais sofisticados estimam a velocidade, comprimento e classe dos veículos, a distância entre eles, sendo capazes de contar veículos com grande precisão. Estes acabam por ser mais utilizados no controlo dos estacionamento, onde têm como principal objectivo contar o número de carros dentro do estacionamento e informar os condutores se o parque está ou não lotado [33]. Na figura 4 está representado um esquema relativo a este sistema.

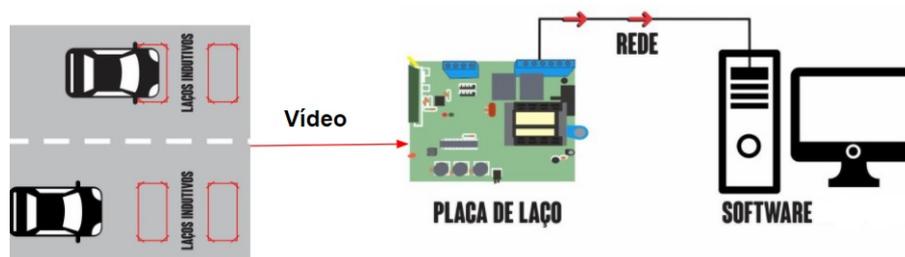


Figura 4: Arquitectura de sensores indutivos (figura adaptada de [14]).

2. Detecção de veículos por vídeo: Os métodos não-intrusivos mais usados recorrem a câmaras de vídeo para recolher de informações do fluxo de tráfego e da detecção de incidentes nas vias. As imagens recolhidas são processadas por um software que permite interpretar os dados e convertê-los em números. A informação recolhida é analisada através de processadores de imagens que usam técnicas de Inteligência Artificial (por exemplo *Computer Vision*), que permitem interpretar as condições de tráfego.

É de referir que, ao combinar as técnicas de detecção de veículos por câmaras de vídeo com a inteligência artificial, pode-se obter um procedimento mais inteligente e preciso para determinar o estado do tráfego, o que permite actuar de uma maneira mais assertiva no controlo do fluxo de tráfego.

### 2.1.3 Aplicações

Actualmente, aplicações de controlo de tráfego inteligentes usando inovação e tecnologia de ponta, têm ganho bastante relevância no que diz respeito ao descongestionamento do tráfego nas grandes cidades, tornando-as cada vez mais sustentáveis. Nesta secção são mencionadas algumas destas aplicações usadas no contexto rodoviário, sendo que

algumas não são necessariamente dirigidas para otimizar o fluxo de tráfego, mas sim para detecção de veículos de emergência.

1. Sistemas de notificação de veículos em situação de emergência: O sistema de chamada automática, implementado em alguns veículos em caso de acidente, permite estabelecer uma ligação com os serviços de emergência. Este sistema é activado quando o condutor pressiona o botão “eCall” que se encontra no interior da sua viatura, sendo estabelecida uma ligação de emergência para o posto mais próximo, com as devidas informações de localização do veículo, de modo a executar o melhor trajecto e permitir o deslocamento da ajuda sem interrupções [16]. Em Portugal, este sistema já é uma realidade desde 2017 [11]. A figura 5 ajuda a perceber o processo utilizado neste sistema eCall.



Figura 5: Arquitectura do sistema eCall (figura adaptada de [11]).

2. Sistemas com limites de velocidade variáveis: Estes sistemas permitem aumentar ou diminuir a velocidade autorizada para circulação dos automobilistas, dependendo do congestionamento rodoviário e doutros factores. Normalmente esses limites de velocidade são alterados para tentar diminuir o congestionamento do tráfego em situações excepcionais [17];
3. Sistemas com sequência de semáforos dinâmicos: Este tipo de sistemas evita problemas que geralmente surgem quando se usam processamento de imagens e técnicas de interrupção de tráfego. Esta técnica consiste no desenvolvimento de um algoritmo dinâmico de sequência de semáforos usando a tecnologia RFID (do inglês, *Radio Frequency Identification*) em conjunto com bases de dados de tráfego. Este método é aplicado a uma área com várias intersecções e que possui múltiplos veículos. Em geral, esta técnica fornece um esquema eficiente do controlo de tempo dos vários semáforos existentes. O resultado da aplicação desta técnica é um cronograma dinâmico que ajuda a escoar mais rapidamente o trânsito [2].

Perante a análise dos diferentes métodos para detecção de tráfego, pode-se assumir que o uso das câmaras é o mecanismo mais aconselhado para recolher dados sobre o fluxo do tráfego nas vias.

## 2.2 Métodos para controlo de fluxo de tráfego

Uma vez conhecidos alguns dos métodos de recolha de dados de tráfego, é necessário saber como os gerir de modo a promover soluções de ordenamento de tráfego, para prevenir congestionamentos, atrasos, aumento do consumo de combustível, stress dos automobilistas, entre outros factores. As soluções a propor deverão ser óptimas, no sentido de evitar a existência de possíveis congestionamentos e minimizar os custos económicos e ambientais.

As grandes cidades já possuem sistemas de controlo de fluxo do tráfego, cada um configurado com as características de cada cidade [28]. Apresentam-se nas subsecções seguintes alguns sistemas de controlo de fluxo do tráfego.

### 2.2.1 Sistema de Controlo de Tráfego SCOOT

□ SCOOT (do inglês, *Split Cycle and Offset Optimisation Technique*) é um sistema de controlo de tráfego em tempo real, desenvolvido em 1973 pelo Laboratório de Transportes do Reino Unido, tendo sido posto em prática em 1979. Este sistema age de forma inteligente e continua mediante as alterações do fluxo de tráfego ao longo de um dia. São inúmeras as vantagens ao utilizar este mecanismo, desde a organização do congestionamento personalizado, prioridade de transporte público e pedestre, previsão das emissões de gases dos veículos, detecção de incidentes na via, entre outros.

Para o seu funcionamento, este sistema requer um processador central que contém o *Kernel* do SCOOT, usado pela empresa de controlo de tráfego urbano de cada cidade, e o software que controla as comunicações dos equipamentos que estão instalados nas estradas.

O processador e terminais associados em rede são instalados numa sala de controlo de tráfego. As informações sobre o desenho físico das estradas e a forma como os sinais de trânsito controlam os fluxos de tráfego são armazenados em base de dados. É possível aceder ao sistema na rua através de terminais móveis, os quais podem ser usados para inicialização, afinação e manutenção do sistema.

Um sistema destes depende sempre da boa detecção das condições em tempo real do fluxo de tráfego, de maneira a permitir uma resposta rápida e eficaz. Os detectores indutivos são os mais usados, embora possam ser aplicados outros. A localização dos detectores é importante, instalados normalmente em cada ligação das estradas, para conseguir fornecer boas informações sobre o tráfego antes dos veículos chegarem ao ponto das linhas do SCOOT. O sistema consegue detectar os veículos que se encontram nos vários cruzamentos controlados, e, alterar assim o estado dos semáforos em função do número de viaturas.

Desta forma, quando os veículos passam pelo detector, o SCOOT recebe as informações e converte esses dados nas suas unidades centrais e constrói perfis cíclicos de fluxo para cada ligação. Este sistema permite minimizar as fases de verde desnecessárias de maneira a que o trânsito flua com mais eficiência [28]. Na figura 6 está representada a arquitectura do sistema SCOOT.

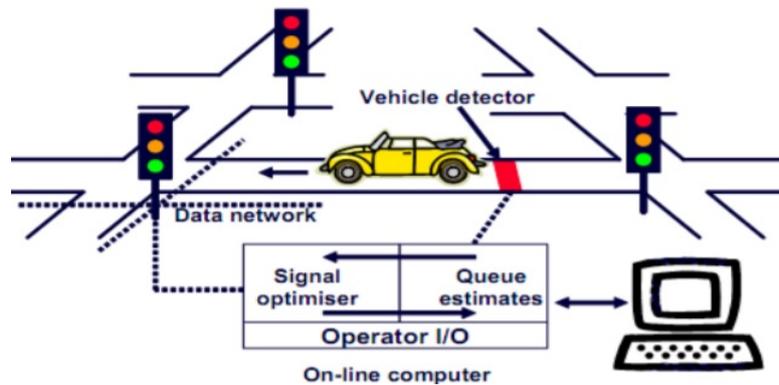


Figura 6: Arquitectura do sistema SCOOT (figura adaptada de [28]).

### 2.2.2 Sistema de Controlo de Tráfego GERTRUDE

O sistema GERTRUDE (Gestão Electrónica de Regulação do Tráfego Rodoviário Urbano Desafiando os Engarrafamentos) é um conjunto de softwares de tráfego, que através dos sensores instalados nas vias da cidade consegue monitorizar em tempo real, o tráfego. Este sistema está adaptado às características do fluxo do tráfego da cidade de Lisboa e os objectivos são: assegurar a fluidez do tráfego, melhorar o desempenho dos transportes colectivos, dar prioridade a veículos de emergência e controlar a poluição atmosférica.

A estratégia de controlo deste sistema divide-se em duas etapas.

Na primeira etapa, é escolhido em tempo real o software de tráfego mais apropriado. O sistema determina o ciclo óptimo para cada zona, tendo por base os coeficientes característicos de cada via. O tempo do ciclo anterior, bem como as informações dos sensores de saturação e contagens, em tempo real, servirão de base para a adaptação do software de maneira a escoar o tráfego de um modo inteligente, minimizando o tempo de espera dos automobilistas.

Na segunda etapa, o software é adaptado às necessidades de cada evento, ou seja, em função de parâmetros/-variáveis como os detectores de filas de espera, prioridade dos transportes colectivos e veículos de emergência [7]. Na figura 7 está representado o esquema de funcionamento do sistema GERTRUDE.

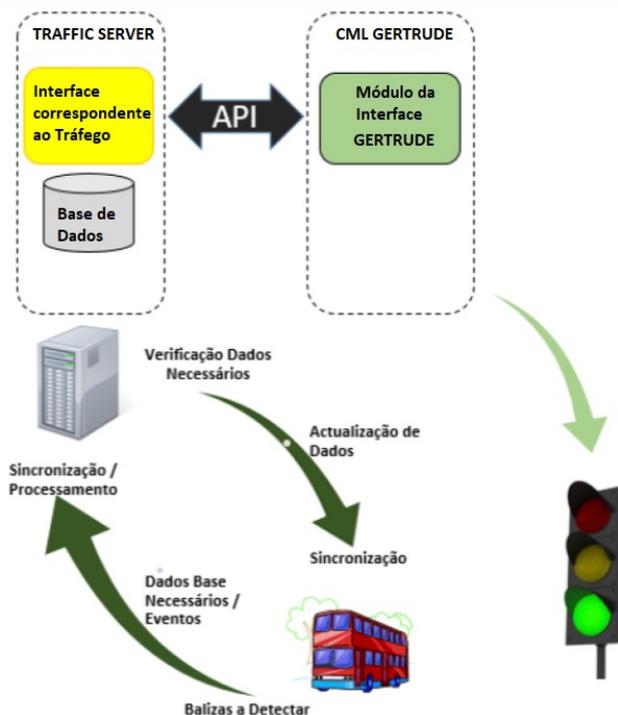


Figura 7: Arquitectura do sistema GERTRUDE (figura adaptada de [9]).

### 2.2.3 Sistema de Controlo de Tráfego UTMS

O UTMS (*Universal Traffic Management Systems*) é um sistema de controlo de tráfego usado no Japão, sendo implementado para melhorar os mecanismos de segurança rodoviária. Este sistema começa por usar vários sensores para recolher os dados do tráfego. De seguida é usado um software para processar a informação recolhida do tráfego e as áreas abrangentes, aperfeiçoando assim o controlo dos sinais dos semáforos e providenciando informações em tempo real. Para automatizar o desempenho do controlo dos semáforos da área coordenada, determina o melhor parâmetro para o controlo do semáforo baseado no volume do tráfego [21]. Estes sistemas são compostos por várias funcionalidades, destacando-se, por exemplo o ICTS (*Integrated Traffic Control Systems*).

Uma vez que o ICTS é a componente principal do UTMS, permite que exista uma gestão do tráfego mais uniforme e segura. Este sistema utiliza tecnologias como sensores infravermelhos e outros softwares para controlar os semáforos e, conseqüentemente lidar com as constantes mudanças de fluxos no tráfego, fornecendo informações em tempo real de tráfego e execução de cada subsistema do UTMS. Uma vez mais, os objectivos são semelhantes aos dos sistemas apresentados anteriormente, nomeadamente: garantir a segurança, permitir um bom fluxo de tráfego, reduzir o congestionamento, reduzir o tempo de viagem, e minimizar os custos económicos e ambientais.

## 2.3 Parâmetros, Variáveis e Modelação

A Engenharia de Tráfego refere-se à análise do comportamento do tráfego e ao seu *design* tendo como objectivo uma mobilidade sustentável e socialmente inclusiva. Os parâmetros e as variáveis do fluxo de tráfego visam fornecer informações sobre a natureza do trânsito, permitindo detectar alterações do seu comportamento.

Nesta secção, apresentam-se alguns tipos de modelações existentes bem como alguns parâmetros e as variáveis mais importantes.

O volume de tráfego é influenciado pelas características individuais de cada veículo e do seu condutor, bem como da forma como os diversos automobilistas interagem entre si.

Na Engenharia de Tráfego é possível assumir que muitas das alterações que podem ocorrer nas vias estão dentro de certos intervalos previstos.

Por exemplo, se a velocidade máxima permitida numa estrada é de 50 km/h para automóveis ligeiros, pode-se pressupor que todo o fluxo de tráfego se mova a uma velocidade média de 40 km/h em vez de 100 km/h ou 20 km/h.

É possível classificar os modelos de tráfego quanto ao seu nível de detalhe, nomeadamente: modelação microscópica, macroscópica e mesoscópica [8].

### 2.3.1 Modelação Microscópica

A modelação microscópica é baseada nas características dos vários movimentos dos veículos de forma individual, com base em vários modelos relacionados com os comportamentos adoptados pelos automobilistas, como por exemplo o *car-following* e o *lane changing* [32]. Em particular, os veículos incorporados na rede são expostos a eventos imprevisíveis que influenciam as atitudes tomadas pelos diversos automobilistas. A posição de cada veículo é acompanhada e actualizada no espaço e no tempo, em função das condições envolventes. Este tipo de modelação permite obter dados como por exemplo: densidade, velocidade, tempo de viagem, níveis de poluição e consumo de combustível.

Uma vantagem deste tipo de modelação é o facto de serem fornecidas muitas informações sobre os recursos do fluxo de tráfego. Porém, esta modelação exige um elevado esforço computacional, principalmente em grandes redes com elevado fluxo de tráfego.

Em particular, apresenta-se de seguida uma breve descrição sobre o modelo *car-following*.

O modelo *car-following* é baseado na ideia de que cada automobilista controla o seu veículo de acordo com os estímulos que recebe do veículo que segue à sua frente. Cada condutor pode responder às condições de tráfego envolvente apenas acelerando ou desacelerando o veículo.

Este modelo pode ser descrito matematicamente com base na relação entre dois veículos consecutivos e no modo como cada condutor reage a um estímulo vindo do veículo que segue à sua frente. Este estímulo pode ser traduzido na forma de aceleração ou de desaceleração do veículo. A resposta de um veículo ao estímulo do *n-ésimo* veículo pode ser traduzida por

$$[Resposta]_n \propto [Estimulo]_n \quad (1)$$

na qual o símbolo " $\propto$ " traduz a ideia de que a resposta é proporcional ao estímulo [27].

Em geral, admite-se que um condutor tenta respeitar uma distância de segurança do seu antecessor de acordo com a velocidade. Pode-se considerar que um estímulo pode ser composto pela velocidade do veículo, velocidades relativas e a distância percorrida. O modelo de *car-following* é ilustrado na figura 8.

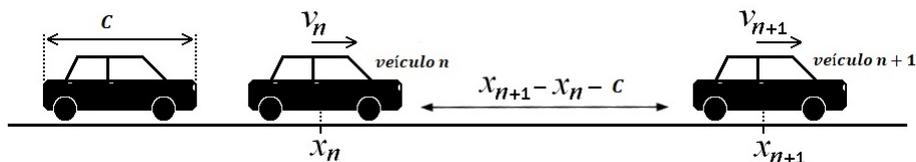


Figura 8: Modelo car-Following (figura adaptada de [4]).  $c$  é o comprimento do veículo,  $x_n$  a sua posição e  $v_n$  a sua velocidade.

Na figura 8 consegue-se ter uma melhor percepção das condições de *car-following*. Nesta figura, o veículo  $n$  representa o veículo em análise e o  $n+1$  representa o veículo que segue à sua frente, designado na literatura por veículo líder [4, 15].

### 2.3.2 Modelação Macroscópica

Na modelação macroscópica o fluxo de tráfego é medido tendo em conta a velocidade e a densidade de tráfego. Este tipo de modelação avalia o fluxo de tráfego como um todo sem considerar as características individuais dos automobilistas.

Os modelos macroscópicos são modelos determinísticos. Modelam o movimento de grupos de veículos, como o comportamento médio de todos os veículos numa dada área num dado período de tempo. Dado que são determinísticos, não existe aleatoriedade nos sistemas rodoviários que o implementam, havendo uma relação determinística entre os *inputs* e os *outputs* do modelo. Por exemplo, caso se realizem duas simulações com este modelo, nas mesmas condições, os resultados obtidos em ambas as simulações serão iguais [6].

Comparativamente ao modelo microscópico, a implementação deste modelo exige um menor esforço computacional. Porém, este modelo não permite analisar de modo tão minucioso as condições de tráfego existentes.

O primeiro modelo macroscópico foi desenvolvido em 1955, por *Lighthill, Whitham e Richards* (LWR), o qual se baseia na equação dada por:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial x} = 0, \quad (2)$$

onde,  $\rho$  representa a densidade de tráfego e  $v$  representa a velocidade média na via. [18].

### 2.3.3 Modelação Mesoscópica

A modelação mesoscópica situa-se entre a modelação microscópica e a modelação macroscópica. Descreve o fluxo de tráfego agregando veículos que possuem características idênticas e fazendo uma abordagem em termos de funções de distribuição. Pode ser aplicada a uma rede rodoviária para modelar o fluxo de tráfego ao longo do tempo. Tem por base um conjunto de veículos que possuem características idênticas entre si. Esta pode ser aplicada a uma rede rodoviária para modelar o fluxo de tráfego ao longo do tempo.

É possível definir uma descrição que combine a modelação mesoscópica com uma modelação macroscópica e, assim sendo, escolher o nível de detalhe pretendido (centralizado apenas no veículo de forma individual ou englobando toda a rede que se queira representar). É de referir que a simulação mesoscópica veio colmatar algumas lacunas das modelações microscópica e macroscópica [39]. A figura 9 exemplifica a organização dos 3 tipos de modelação mencionados.

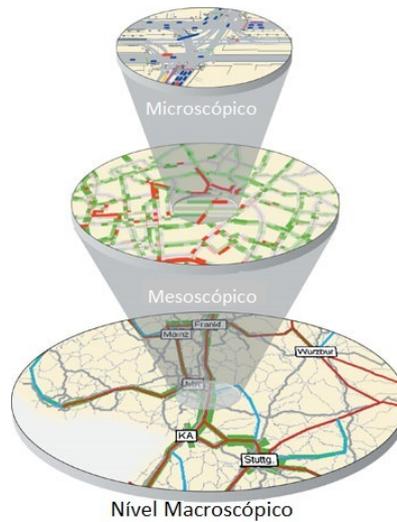


Figura 9: Níveis dos três tipos de modelação (figura adaptada de [23]).

### 2.3.4 Parâmetros e Variáveis Fundamentais

Para a regulação dos semáforos é necessário fixar e variar tempos de duração para cada uma das cores dos semáforos. Assim, é importante introduzir algumas definições e notações para definir um modelo de fluxo de tráfego.

- Intervalo: Um intervalo é o tempo atribuído a cada cor da sequência do semáforo.
- Ciclo: Um ciclo corresponde ao tempo, em segundos, que é necessário para completar todas as sequências de cores do semáforo. Quanto maior o tempo de ciclo, maior o tempo médio de espera do cruzamento.
- Fase: Uma fase representa um intervalo de verde para uma faixa do cruzamento, durante a qual os veículos dessa faixa podem avançar no cruzamento, enquanto outros veículos ficam impedidos de avançar por estarem perante um intervalo de cor vermelha. Assim, durante uma fase, não existem movimentos que gerem conflitos entre veículos, permitindo que o trânsito flua e apenas seja interrompido com segurança antes do início da próxima fase. Todas estas configurações das luzes do semáforo manter-se-ão de forma cíclica.

O diagrama de tempos, representado na figura 10, mostra os intervalos de tempo de ciclo para um cruzamento com quatro faixas, em que cada faixa tem um semáforo, daí existirem 4 fases ilustradas na figura.



Figura 10: Esquema de um ciclo com 4 fases.

De acordo com [26] existem muitas variáveis que deverão ser tidas em consideração na modelação do tráfego, nomeadamente: fluxo de saturação, capacidade de um cruzamento/intersecção, tempo de verde, grau de saturação e tempo médio de atraso dos veículos. Os valores destas variáveis podem ser calculadas do seguinte modo:

- *Fluxo de Saturação (S)*: corresponde ao fluxo máximo de veículos que podem atravessar uma determinada intersecção, assumindo-se as condições ideais na via. A fórmula para o cálculo do valor de  $S$  é dada por [19]:

$$S = \frac{3600}{H} \quad (3)$$

onde  $S$  expressa o número de veículos que passam numa hora e  $H$  é o tempo em segundos que um carro leva a atravessar o cruzamento (em unidades de segundo/veículo).



Figura 11: Gráfico do Fluxo de Saturação (figura adaptada de [36]).

É de salientar que, numa fase, o tempo de verde nunca é totalmente aproveitado, pois existe uma perda de tempo no início desta fase, devido à reacção dos automobilistas (ver figura 11). À medida que o tempo passa, a taxa de escoamento vai aumentando, e quando atinge um valor máximo, esse valor será o fluxo de saturação. Importa realçar que este volume só estabiliza após os primeiros 10 segundos do tempo de verde, ou seja, na passagem do quarto veículo, como é reportado na literatura [36].

De modo a maximizar a passagem de veículos no tempo de verde foram implementados nos semáforos temporizadores, para informar o automobilista do tempo que falta para a transição para o próximo estado do semáforo (verde ou vermelho). Deste modo, os condutores encaram as duas situações de forma diferente, adiando a imobilização quando vai surgir o vermelho e antecipando o arranque quando surgir o verde, o que implica que o tempo perdido de verde seja menor.

- *Tempo de Verde Efectivo ( $g_{ef}$ )*: é dado pela diferença entre o tempo de verde normal ( $g$ ), e o tempo perdido por fase ( $I_i$ ):

$$g_{ef} = g - I_i \quad (4)$$

- *Capacidade do Cruzamento ( $C_p$ )*: dá o número máximo de veículos capazes de atravessar um cruzamento durante um ciclo. A capacidade do cruzamento ( $C_p$ ) é dada por:

$$C_p = S \left( \frac{g_{ef}}{C} \right) \quad (5)$$

onde  $C_p$  expressa o número máximo de veículos que passam numa hora,  $S$  é o fluxo de saturação,  $g_{ef}$  é o tempo verde efectivo e  $C$  o tempo de ciclo.

- *Número de Paragens ( $P$ )*: dá o número de vezes que um veículo fica retido na faixa de trânsito, desde que chega aquela faixa até passar o cruzamento. O seu valor é dado por:

$$P = \left( \frac{1 - \frac{g}{C}}{1 - \frac{q}{S}} \right) \quad (6)$$

onde  $g$  é o tempo de verde normal,  $C$  é o tempo de Ciclo,  $q$  representa o fluxo de veículos que estão em espera na faixa e  $S$  é fluxo de saturação.

É de salientar que, quando é realizado um estudo do tráfego num dado cruzamento ou numa rede de cruzamentos, muitas são as variáveis que interferem na dinâmica das condições das vias. No capítulo 3 é descrito o cruzamento desenvolvido, bem como as condições e as variáveis que são necessárias para a implementação de um sistema de tráfego muito próximo da realidade.

### 3 Simulador Convencional

Neste capítulo é descrito um simulador convencional, para controlo do fluxo de tráfego num cruzamento com quatro faixas, desenvolvido no âmbito desta dissertação.

#### 3.1 Descrição do Simulador Convencional

O simulador aqui proposto permite modelar o comportamento de cada veículo durante todo o tempo de simulação.

Para o desenvolvimento deste simulador usou-se a biblioteca *p5.js* [38] *open-source* escrita na linguagem *JavaScript*, e programação em HTML e CSS. Com o auxílio destas ferramentas de programação, foi possível desenvolver uma interface gráfica dinâmica, que representa o cruzamento bem como o fluxo de tráfego, em estudo.

Este simulador foi desenvolvido tendo por base os seguintes pressupostos:

- Cruzamento com quatro faixas de rodagem (denotadas por A, B, C, D), as quais podem ser constituídas por várias vias de trânsito (Norte - Sul, Sul - Norte, Este - Oeste e Oeste - Este). Notar que uma faixa pode ser constituída por várias vias (ver Figura 12);

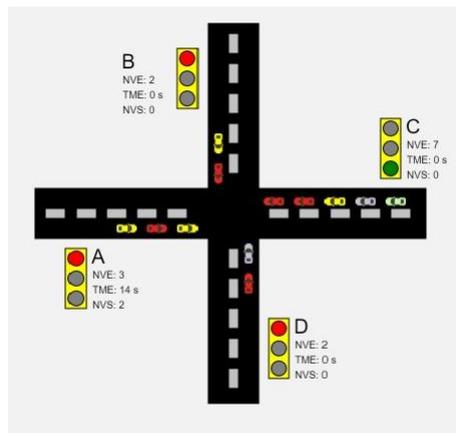


Figura 12: Faixas A, B, C e D.

- Se o semáforo está verde, então os veículos dessa faixa podem seguir em qualquer direcção;
- Cada semáforo tem apenas duas cores (verde e vermelho), cada uma com um tempo fixo de 30 segundos. Neste simulador não foi considerada a luz amarela;
- Para cada faixa existe um processo aleatório para gerar os veículos. Por defeito, por forma a gerar aleatoriedade, a cada 4 segundos é colocado um veículo numa das faixas, A, B, C ou D.

No desenvolvimento deste simulador considerou-se apenas veículos ligeiros. No entanto, o simulador poderá ser facilmente adaptado para outro tipo de veículos (pesados, transporte colectivo de passageiros), e também para determinados comportamentos específicos dos automobilistas. Um dos objectivos deste simulador é calcular o tempo médio de espera dos veículos em cada faixa, e o tempo médio de espera em todo o cruzamento. A recolha dos dados de fluxo de tráfego fornecidos por este simulador é crucial para posteriormente comparar estes dados com os obtidos através de um simulador inteligente proposto nesta dissertação no capítulo 4.

## 3.2 Variáveis do fluxo de tráfego

Como já referido, existem várias variáveis que interferem na dinâmica do fluxo de tráfego. Para simular cenários de fluxo de tráfego muito próximos da realidade, foi necessário definir algumas variáveis fundamentais, pois um sistema de tráfego é um sistema bastante complexo caracterizado pela aleatoriedade e imprevisibilidade das condições que nele operam a cada instante. Apresenta-se de seguida as várias variáveis que interferem no fluxo de tráfego.

Um dos objectivos desta secção é desenvolver uma fórmula matemática que quantifique quanto tempo demora cada veículo (que se encontra numa das quatro faixas) a percorrer a distância desde a posição em que se encontra até à passagem pelo semáforo.

### 3.2.1 Tempo de Reacção ( $T_{reac}$ )

Na condução, o tempo de reacção é definido como sendo o intervalo de tempo entre o instante da identificação de um estímulo ou evento imprevisto e o momento em que o condutor inicia a resposta a esse estímulo. Embora os condutores tenham a sensação de reagir rapidamente, de facto, entre *avistar* o obstáculo e o *actuar* decorre, num condutor em condições normais, o tempo varia aproximadamente entre 1.5 segundos e 2 segundos [35]. No desenvolvimento do simulador assumiu-se que  $T_{reac}$  será gerado aleatoriamente no intervalo [1.5s, 2s].

De notar que este é o tempo necessário para detectar uma situação irregular através dos sentidos (na condução o sentido mais utilizado é a visão e depois a audição), identificá-la, analisá-la e decidir qual a resposta mais adequada e o início da concretização dessa resposta [24].

### 3.2.2 Velocidade Média ( $V_{med}$ )

A velocidade média de uma viatura é dada pelo quociente entre a distância percorrida e o tempo que demora a percorrer essa distância. No desenvolvimento do simulador assumiu-se que, após os 10 primeiros segundos de tempo de verde, os veículos movimentam-se a uma velocidade média de 12m/s [22].

### 3.2.3 Distância Bumper-to-Bumper ( $D_{btb}$ )

Na segurança rodoviária a distância de segurança é bastante importante, o que leva a ter em consideração esta variável no desenvolvimento do simulador. A distância *bumper-to-bumper* consiste na distância entre o pára-choques frontal e o pára-choques traseiro, como se pode ver na figura 13. Quando o semáforo está com a cor vermelha, a maioria dos automobilistas tende a manter uma distância muito pequena ao carro que se encontra à sua frente, deixando apenas alguns centímetros entre o seu pára-choques frontal e pára-choques traseiro da viatura que segue imediatamente à sua frente.



Figura 13: Distância *bumper-to-bumper*.

É de referir que, quanto menor a distância  $D_{btb}$ , mais rápido será o escoamento da fila de espera, minimizando o atraso. Para ter em consideração as diferentes atitudes de cada automobilista, o valor desta variável será gerado aleatoriamente no intervalo  $[0.38m, 1.8m]$  como sugerido na literatura [11] (ver Figura 14).

Assim,  $D_{btb} = Random_{Dec}([0.38, 1.8])$  onde  $Random_{Dec}$  é a função que gera um valor real no intervalo dado.

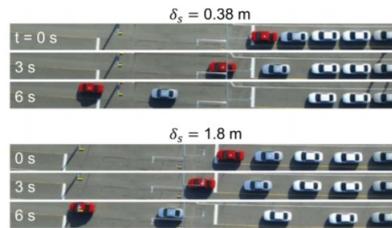


Figura 14: Distâncias *bumper-to-bumper* consideradas (figura adaptada de [11]).

### 3.2.4 Comprimento total de um veículo ( $C_{tvei}$ )

No desenvolvimento do simulador considerou-se que todos os veículos que chegam ao cruzamento são veículos do tipo ligeiros de passageiros, com um comprimento de 5 ou 6 metros. Na prática, este comprimento pode ser captado por sensores.

No desenvolvimento do simulador assumiu-se que o comprimento total do veículo tem em conta o seu comprimento, gerado aleatoriamente pela função  $Random_{Int}([5, 6])$ , mais a distância  $D_{btb}$ :

$$C_{tvei} = Random_{Int}([5, 6]) + D_{btb} \quad (7)$$

### 3.2.5 Distância do veículo ao semáforo ( $D_{veisem}$ )

A distância de um determinado veículo ao semáforo ( $D_{veisem}$ ) é dada pela soma dos comprimentos  $C_{tvei}$  dos veículos que se encontram à sua frente.

### 3.2.6 Tempos estimados ( $T_{verini}$ , $T_{fv}$ , $T_{tdeco}$ )

Como já referido, o tempo de verde ( $T_{tver}$ ) está fixo em 30 segundos. É de salientar que nos primeiros 10 segundos, devido ao tempo de reacção e a outros factores que podem interferir na atenção dos automobilistas, assumiu-se que apenas conseguem passar o cruzamento, nestes 10 segundos, 3 ou 4 veículos (denotado pela variável  $N_{veitverini}$ ) [36]. Consequentemente, pode-se dividir o tempo de verde em dois subintervalos:

- Tempo de verde inicial ( $T_{verini}$ ): corresponde aos primeiros 10 segundos do tempo total de verde;
- Tempo de fluxo variável ( $T_{fv}$ ): é dado pelo tempo total de verde menos  $T_{verini}$ ,

$$T_{fv} = T_{tver} - T_{verini}. \quad (8)$$

Para determinar o tempo de reacção dos automobilistas que estão na faixa (após os primeiros 10 segundos) que conseguem ainda passar o cruzamento (tempo decorrido -  $T_{tdeco}$ ) é proposta a seguinte fórmula:

$$T_{tdeco}(P_{vei}) = (P_{vei} - N_{veitverini}) \times T_{reac} + T_{verini}, \quad (9)$$

onde  $P_{vei}$  é a posição do veículo na fila de espera ( $P_{vei} > N_{veitverini}$ ),  $N_{veitverini}$  é o número de veículos que passam nos primeiros 10 segundos,  $T_{reac}$  é o tempo de reacção do condutor e  $T_{verini}$  representa os 10 segundos iniciais do tempo de verde.

A tabela seguinte sumariza as variáveis que foram tidas em consideração no desenvolvimento do simulador.

Tabela 1: Variáveis e expressões usadas na fórmula.

Variáveis	Definição	Expressão Matemática
$C_{vei}$	Comprimento de cada veículo	$\text{Random}_{Int}([5, 6])$
$V_{med}$	Velocidade média	12 m/s
$D_{btb}$	Distância bumper-to-bumper	$\text{Random}_{Dec}([0.38, 1.8])$
$C_{tvei}$	Comprimento total de cada veículo	$C_{tvei} = C_{vei} + D_{btb}$
$T_{reac}$	Tempo de reacção	$\text{Random}_{Dec}([1.5, 2])$
$T_{tver}$	Tempo de verde	30 s
$T_{verini}$	Primeiros 10 s do tempo total de verde	10 s
$N_{veitverini}$	Nº de veículos nos primeiros 10 s	$\text{Random}_{Int}([3, 4])$
$T_{fv}$	Tempo de fluxo variável	$T_{fv} = T_{tver} - T_{verini}$
$P_{vei}$	Posição do veículo na fila de espera	1,2,...
$D_{veisem}$	Distância do veículo ao semáforo	$D_{veisem} = (P_{vei} - 1) \times C_{tvei}$
$T_{tdeco}$	Tempo de reacção do veículo antes do arranque	$T_{tdeco} = (P_{vei} - N_{veitverini}) \times T_{reac} + T_{verini}$

Tendo em conta as variáveis acima definidas, foi possível desenvolver uma função  $f$ , que quantifica o tempo que um veículo, numa determinada posição  $P_{vei}$  (para  $P_{vei} > N_{veitverini}$ ), demora a transpor o cruzamento. A função proposta é definida por:

$$f(P_{vei}) = \frac{(P_{vei} - 1) \times C_{tvei}}{V_{med}} + T_{reac} + T_{tdeco}(P_{vei}). \quad (10)$$

Notar que, o veículo na posição  $P_{vei}$  só passa o cruzamento se  $f(P_{vei}) \leq 30$ , uma vez que o tempo de verde fixo é de 30 segundos.

Apresenta-se a seguir um exemplo da aplicação da função  $f$  definida em (10).

- Suponhamos que temos uma faixa com 12 carros (ver figura 15) em espera. Pretende-se saber se o veículo 11 consegue transpor o cruzamento enquanto o semáforo está verde (30 segundos). Considerando o carro na posição 11, e que  $C_{tvei} = 6.8m$ ,  $T_{reac} = 1.5s$  e  $N_{veitverini} = 4$ , o tempo que este veículo demora a transpor o cruzamento é:

$$f(11) = \frac{(11 - 1) \times 6.8}{12} + 1.5 + ((11 - 4) \times 1.5) + 10 = 27s. \quad (11)$$

Como  $f(11) = 27 \leq 30$ , confirma-se que este veículo ainda conseguiu transpor o cruzamento enquanto o semáforo está verde. Para os veículos em determinadas posições,  $P_{vei}$ , e para os quais  $f(P_{vei}) > 30$  significa que esses veículos não conseguiram transpor o cruzamento. Este tipo de informação também é importante uma vez que permite determinar o número de veículos que ficaram retidos na faixa.

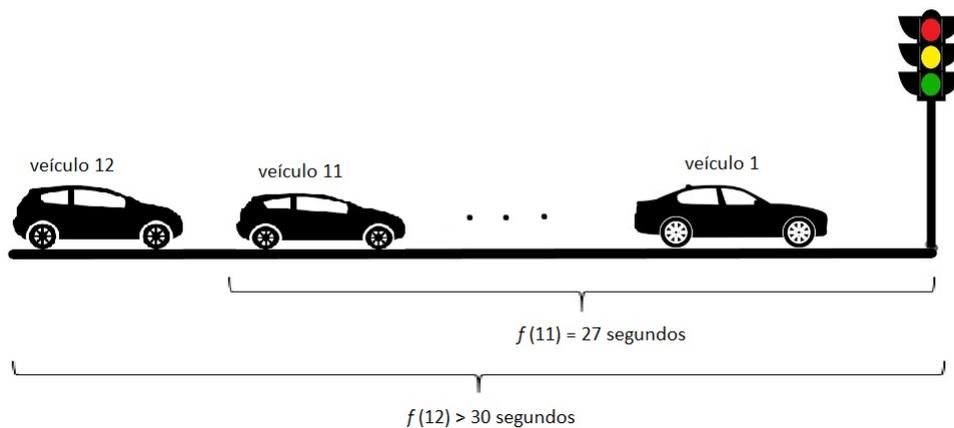


Figura 15: Esquema exemplificativo da aplicação da função  $f$  definida em (10).

Para finalizar é de salientar a importância de incorporar as quatro variáveis aleatórias, mencionadas acima, no desenvolvimento do simulador. Estas variáveis vão permitir gerar/criar cenários de simulação mais próximos da realidade, e aumentar a diversidade dos cenários gerados.

Além disso, foi também criada uma função aleatória para gerar os veículos para as quatro faixas do cruzamento. Esta função gera/injecta aleatoriamente 15 veículos por minuto, onde de 4 em 4 segundos é gerado um veículo que pode chegar à faixa A, B, C ou D.

O simulador convencional desenvolvido tem um menu na sua interface gráfica que permite alterar alguns parâmetros mediante o cenário de simulação em estudo, nomeadamente: tempo de verde ( $T_{ver}$ ), velocidade média com que os veículos transpõem o cruzamento ( $V_{med}$ ), tempo de simulação e o fluxo máximo de veículos para o cruzamento por minuto (ver figura 16).

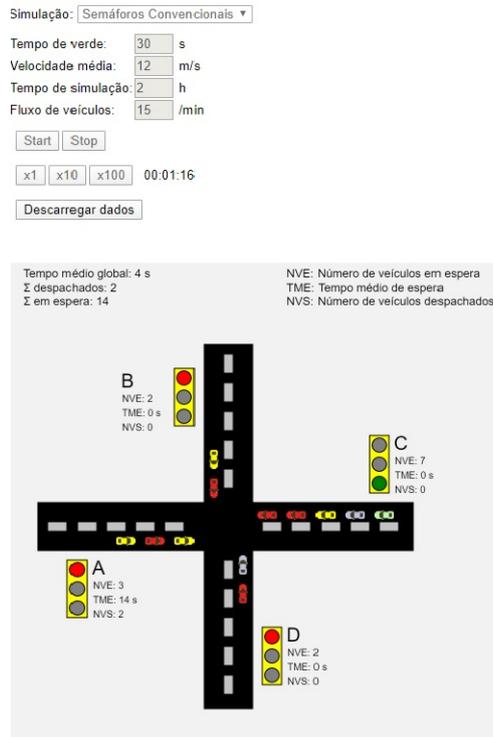


Figura 16: Interface gráfica do simulador.

O simulador desenvolvido permite conhecer a dinâmica do trânsito num cruzamento. Os outputs do simulador no final de cada simulação, para cada faixa, são: *Tempo médio de espera* ( $T_{med}$ ), *Maior tempo de espera* ( $T_{max}$ ), *Média de paragens* ( $P_{med}$ ), *Maior número de paragens* ( $P_{max}$ ) e o *Número de veículos despachados* ( $N_{vd}$ ). Destes outputs, o tempo médio de espera dos veículos é aquele que se pretende minimizar neste trabalho.

O cálculo do  $T_{med}$ , por faixa, tem em conta a hora de chegada e a hora de partida de cada veículo, e o número de veículos que se encontram na faixa.

Seja  $t_1, \dots, t_n$  o conjunto de tempos de espera dos veículos que foram despachados durante um ciclo, numa dada faixa. O tempo médio de espera destes veículos nessa faixa é dado por:

$$T_{med} = \frac{1}{n} \left( \sum_{k=1}^n t_k \right) \quad (12)$$

onde  $n$  é o número de veículos despachados, num ciclo.

O símbolo  $N_{vd}$  indica o número de veículos que conseguiram atravessar a intersecção quando a luz verde estava activa, numa dada faixa, durante a simulação. Se substituir o  $n$  por  $N_{vd}$  na equação (12), obtêm-se o tempo médio de espera, por faixa, durante o período de simulação.

O símbolo  $T_{max}$  indica o maior tempo médio de espera, numa faixa, durante a simulação.

O cálculo da *Média de paragens*, por faixa, tem em conta se o semáforo está verde ou vermelho para cada veículo. Aquando da chegada de um veículo, se o semáforo está vermelho, o número de paragens é inicializado em 1, caso contrário é inicializado a zero. A seguir, por cada ciclo do semáforo em que o veículo não consegue passar o cruzamento, é contabilizada mais uma paragem. Assim, a média de paragens por faixa, é dada pelo soma das paragens de todos os veículos ( $p_k$ ,  $k = 1, \dots, m$ ), durante a simulação, dividida pelo número de veículos ( $m$ )

nessa faixa:

$$P_{med} = \frac{1}{m} \left( \sum_{k=1}^m p_k \right). \quad (13)$$

O  $P_{max}$  é o número máximo de paragens, numa faixa, durante a simulação.

### 3.3 Resultados com o simulador

Com o objectivo de avaliar a dinâmica do simulador convencional, foi realizada uma simulação com duração de 1 hora, com uma velocidade média de 12 m/s, tempo de verde fixo de 30 s e um fluxo de 15 carros por minuto. Apresenta-se na tabela 2, os resultados obtidas relativos ao cenário de simulação apresentado.

Tabela 2: Resultados da simulação com tempos de verde fixos.

Faixa	$T_{med}$	$T_{max}$	$P_{med}$	$P_{max}$	$N_{vd}$
<b>A</b>	64 s	145 s	0.78	1	209
<b>B</b>	60 s	119 s	0.75	1	187
<b>C</b>	64 s	126 s	0.81	1	247
<b>D</b>	61 s	121 s	0.77	1	248

Da tabela 2 conclui-se que o tempo médio de espera dos veículos,  $T_{med}$ , varia entre os 60 s e 64 s para as quatro faixas. O tempo máximo de espera,  $T_{max}$  é diferente para cada uma das faixas. O tempo de verde fixo (30 s) não foi suficiente para escoar todos os veículos que se encontravam em espera, fazendo aumentar os tempos médios de espera  $T_{med}$  e consequentemente o  $T_{max}$ .

Relativamente ao número médio de paragens em cada faixa, verifica-se que houve um ligeiro aumento na faixa C e consequentemente um maior tempo médio de espera nessa mesma faixa. Relativamente ao número total de veículos despachados durante o tempo de verde,  $N_{vd}$ , este foi ligeiramente superior na faixa D. Estes valores variam de faixa para faixa, uma vez que os factores aleatórios (como por exemplo a injeção de carros) influenciam as condições de tráfego.

## 4 Simulador Inteligente

Os métodos e os algoritmos mais usados para controlo de tráfego são de Optimização Clássica, Optimização Heurística ou de Inteligência Artificial. Em qualquer um destes métodos, o objectivo é maximizar a fluidez do tráfego e consequentemente minimizar o tempo médio de espera dos automobilistas.

O *Machine Learning* (ML) é uma sub-área da Inteligência Artificial que se dedica ao estudo de algoritmos de computação que melhoram automaticamente através da experiência. Os algoritmos de ML constroem um modelo matemático baseado em dados de uma amostra, conhecidos como dados de treino, para fazer previsões ou decisões sem ser explicitamente programado para isso.

Há 50 anos atrás, a técnica de ML ainda era ficção científica. Hoje é parte integrante de nossas vidas, ajudando-nos a fazer tudo desde pesquisar por fotos até conduzir carros. A disciplina de ML evoluiu ao longo do tempo, principalmente devido a filósofos, cineastas, matemáticos e cientistas da computação que alimentaram o sonho de criar máquinas que podem aprender.

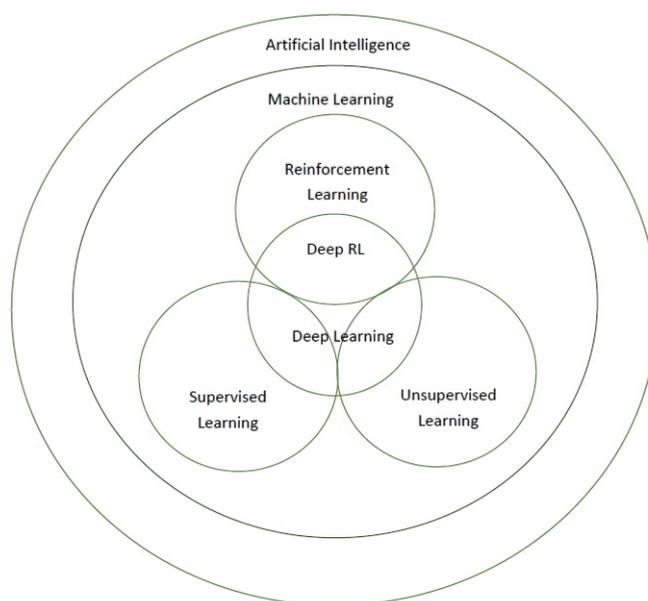


Figura 17: Diagrama das técnicas de Inteligência Artificial (figura adaptada de [31]).

Podem distinguir-se os seguintes trabalhos que constituem marcos na evolução do mundo da computação inteligente: Em 1642 Blaise Pascal (com 19 anos) criou uma máquina aritmética para o seu pai (coletor de impostos). A máquina conseguia adicionar, subtrair, multiplicar e dividir. Três séculos depois, a Autoridade Tributária usa algoritmos de ML para combater a fuga aos impostos. Em 1943 um neurofisiologista e um matemático escreveram um artigo sobre como os neurónios humanos poderiam funcionar. Para ilustrar a teoria, eles modelaram uma rede neuronal com circuitos eléctricos. Nos anos 50, os cientistas da computação começaram a aplicar esta ideia ao seu trabalho (começava a surgir a ideia de uma rede neuronal como um sistema com base no funcionamento do sistema nervoso humano).

A primeira rede neuronal aplicada a um problema do mundo real (1959), o MADALINE de Stanford, usou um filtro adaptável para remover ecos nas linhas telefônicas. Ainda está em uso nos dias de hoje. Em 2012, uma rede neuronal criada pela Google aprendeu a reconhecer humanos e gatos nos vídeos do YouTube - sem precisar de saber como caracterizá-los, e em 2015 o PayPal decidiu combater a fraude e a lavagem de dinheiro adoptando uma abordagem híbrida. Detectives humanos definem as características do comportamento criminoso e, em seguida, um programa de ML usa esses atributos para erradicar os bandidos das plataformas do PayPal.

Pode então concluir-se que os algoritmos de ML estão presentes nas nossas vidas e vieram para ficar. Como Big Data é a tendência mais quente na indústria de tecnologia no momento, os algoritmos de ML são incrivelmente poderosos para fazer previsões ou sugestões calculadas com base em grandes quantidades de dados.

Diversas técnicas de ML foram propostas na literatura ao longo dos anos. O objectivo final é obter um método mais robusto e eficaz, que permita a correcta detecção de padrões com o mínimo custo computacional possível. Os algoritmos de ML podem ser divididos em 3 categorias: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço (ver figura 17). A aprendizagem supervisionada é útil nos casos em que uma propriedade (rótulo) está disponível para um determinado conjunto de dados (conjunto de treino). A aprendizagem não supervisionada é útil nos casos em que o desafio é descobrir relações implícitas num conjunto de dados não rotulados. A aprendizagem por reforço situa-se entre as duas categorias anteriores – existe alguma forma de *feedback* disponível para cada passo ou acção preditiva, mas sem rótulo preciso ou mensagem de erro.

Uma das abordagens mais importante de ML são as Redes Neurais Artificiais (do inglês, *Artificial Neural Networks* (ANN)). Os algoritmos de ML são inspirados na estrutura e nos aspectos funcionais das redes neuronais biológicas. As computações são estruturadas em termos de um grupo interconectado de neurónios artificiais, processando informação usando uma abordagem de conexionismo na computação. As ANN modernas são ferramentas de modelação de dados estatísticos não lineares. Normalmente são usados para modelar relações complexas entre entradas e saídas, para encontrar padrões nos dados, ou para capturar a estrutura estatística em uma distribuição de probabilidade conjunta desconhecida entre variáveis observáveis. Os preços mais baixos dos hardwares e o desenvolvimento de GPUs para uso pessoal nos últimos anos tem contribuído para o desenvolvimento do conceito de *Deep Learning* (DL), que consiste numa ANN com múltiplas camadas ocultas. Esta abordagem tenta modelar a forma com que o cérebro humano processa luz e som na visão e audição. Algumas aplicações úteis do ML profundo são visão computacional e reconhecimento de fala.

No âmbito desta dissertação, com o objectivo de desenvolver um simulador inteligente para controlo do fluxo de tráfego, optou-se por implementar uma ANN, a qual será descrito na secção seguinte.

#### 4.1 Descrição do simulador inteligente

Para o desenvolvimento do simulador inteligente recorreu-se à biblioteca *ML5.js* [37]. Trata-se de uma biblioteca de alto nível, contendo modelos de *Machine Learning* que tem como base a biblioteca *TensorFlow.js*. Nesta dissertação, como já referido, optou-se pelo uso de uma ANN (aprendizagem supervisionada). Para usar este tipo de modelo será necessário construir um conjunto de dados de fluxo de tráfego para o treino da ANN. Entenda-se por treino a parametrização deste modelo de modo a maximizar a fluidez do tráfego e minimizar o tempo de espera dos automobilistas, no cruzamento em estudo.

## 4.2 Conjuntos de dados de treino

Para obter os conjuntos de dados de treino foi necessário desenvolver um simulador manual. O simulador manual tem por base o simulador convencional apresentado no capítulo anterior e implementa um método que permite activar o semáforo manualmente através do teclado do computador. O utilizador actua como se fosse um polícia sinaleiro a olhar para as quatro faixas e a tomar a decisão de qual o semáforo a activar, tendo em conta o número de veículos e o tempo médio de espera, em cada faixa (ver figura 18).



Figura 18: Polícia Sinaleiro na cidade de Braga [10].

Para cada ordem dada pelo polícia sinaleiro (simulador manual-teclado) é guardado num conjunto de dados de treino (ficheiro *.json*), para cada faixa, o número de veículos, os tempos médios de espera, o semáforo que foi activado e o respectivo tempo de verde para essa faixa.

Uma vez que a fórmula (12) que é usada para calcular os tempos médios de espera como uma média simples dos tempos de espera dos veículos despachados não tem em consideração a ordem de chegada dos veículos, alguns veículos podem ficar com tempos médios de espera mais elevados. Para tentar solucionar este problema, procurou-se avaliar o uso de médias pesadas para calcular o tempo médio de espera, por faixa, em cada ciclo.

Seja  $t_1, \dots, t_n$  o conjunto dos tempos de espera dos veículos que foram despachados, numa dada faixa, num ciclo.

Foram propostas as seguintes médias pesadas para calcular os tempos médios de espera dos veículos, por faixa, num ciclo.

Atribuição de um peso de 50% ao primeiro veículo da fila e 50% aos restantes veículos dessa fila.

$$T_{med-50\%vec1} = t_1 \times 0.5 + \frac{1}{n-1} \left( \sum_{k=2}^{n-1} t_i \right) \times 0.5. \quad (14)$$

Atribuição de um peso de 70% ao primeiro veículo da fila e 30% aos restantes veículos dessa fila.

$$T_{med-70\%vec1} = t_1 \times 0.7 + \frac{1}{n-1} \left( \sum_{k=2}^{n-1} t_i \right) \times 0.3. \quad (15)$$

Atribuição de um peso de 70% à primeira metade da fila e 30% à outra metade.

$$T_{med-70\%mf} = \frac{1}{\lceil n/2 \rceil} \left( \sum_{k=1}^{\lceil \frac{n}{2} \rceil} t_i \right) \times 0.7 + \frac{1}{n - \lceil n/2 \rceil} \left( \sum_{k=\lceil \frac{n+2}{2} \rceil}^n t_i \right) \times 0.3, \quad (16)$$

onde  $\lceil y \rceil$  representa a função parte-inteira (*ceiling*) e converte o número real  $y$  no menor inteiro maior ou igual a  $y$ .

Assim, tendo em consideração as fórmulas (12), (14), (15) e (16), foram construídos quatro conjuntos de treino usando o simulador manual-teclado, um para cada fórmula para o calculado o tempo médio de espera, por faixa, em cada ciclo, respectivamente. Estes conjuntos de treino vão ser designados daqui em diante por: DTreino- $T_{med}$ , DTreino- $T_{med-50\%vec1}$ , DTreino- $T_{med-70\%vec1}$ , DTreino- $T_{med-70\%mf}$ .

Para construir cada um destes conjuntos de treino (ficheiros *.json*) consideraram-se simulações de 30 minutos no simulador manual-teclado, ou seja, uma simulação por conjunto de treino a construir, o que totaliza 2 horas de simulação. Durante o período de cada simulação de 30 minutos, foram guardados no respectivo conjunto de treino os dados correspondentes às decisões do “polícia sinaleiro” (simulador manual-teclado).

Ilustra-se na figura 19 um elemento/dado do conjunto de treino DTreino- $T_{med}$ , (ficheiro *.json*). Como se pode verificar, o semáforo activado pelo “polícia sinaleiro” foi o da faixa A, uma vez que possui um maior número de veículos e um maior tempo médio de espera.

```
{
  "numeroCarrosA": 3,
  "numeroCarrosB": 2,
  "numeroCarrosC": 1,
  "numeroCarrosD": 1,
  "TempoEsperaA": 22,
  "TempoEsperaB": 12,
  "TempoEsperaC": 4,
  "TempoEsperaD": 8,
  "Semaforo": "A",
  "TempoVerdeManual": 13
},
```

Figura 19: Parte do *Dataset* para treino da Rede Neuronal.

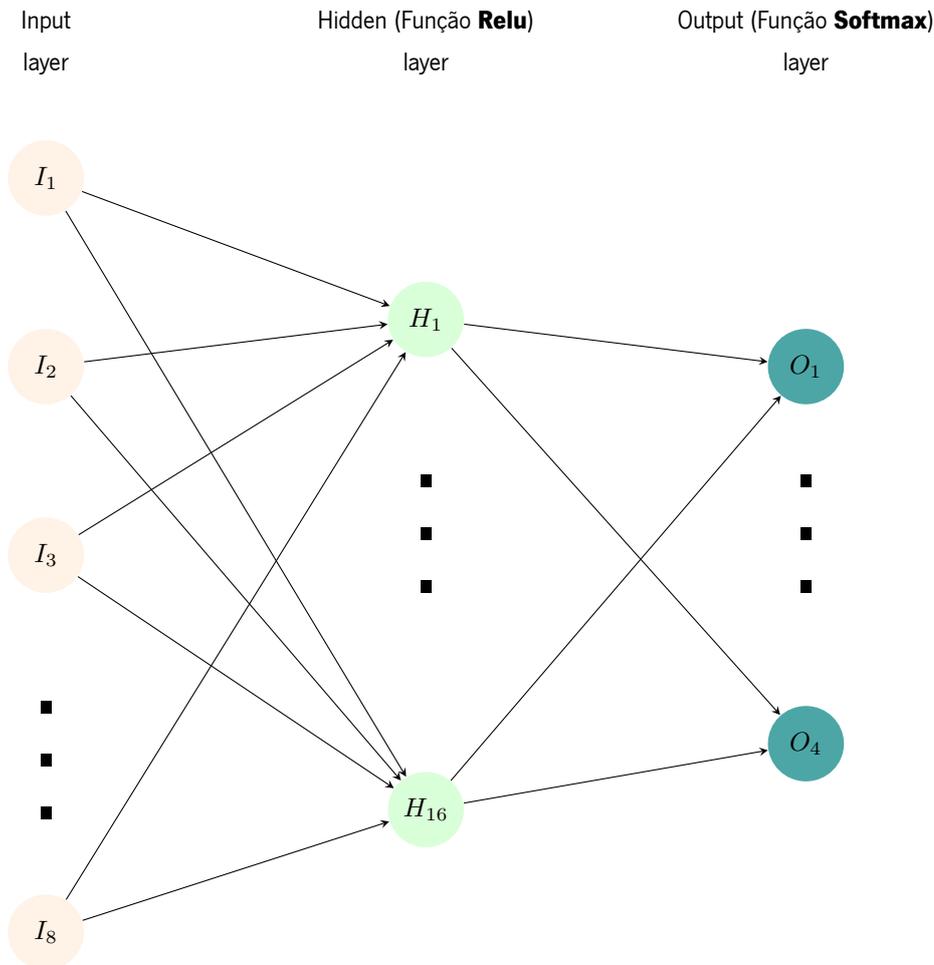
Após a construção dos conjuntos de treino de forma manual, a cardinalidade destes conjuntos foi a seguinte:

- #DTreino- $T_{med}$  = 110
- #DTreino- $T_{med-50\%vec1}$  = 112
- #DTreino- $T_{med-70\%vec1}$  = 107
- #DTreino- $T_{med-70\%mf}$  = 108

Para finalizar é de salientar que, em cada conjunto de treino os atributos são: o número de veículos e o tempo médio de espera, em cada uma das 4 faixas (primeiras 8 linhas da Figura 19), e a label (rótulo) é o semáforo a activar.

### 4.3 Redes Neurais Artificiais usadas

Como já referido, para o desenvolvimento do simulador inteligente, optou-se por usar ANNs. A figura seguinte ilustra a arquitectura das ANNs usadas para o desenvolvimento do simulador inteligente, as quais foram definidas recorrendo à biblioteca *ML5.js*.



Neste diagrama, o significado dos símbolos é o seguinte.

- $I_1$  até  $I_8$  - camada de *input* - corresponde aos dados de entrada da rede, nomeadamente: o número de veículos e o tempo médio de espera, para cada uma das faixas.
- $H_1$  até  $H_{16}$  - camada escondida - camada formada por 16 de neurónios, e em que cada neurónio tem como função de activação a função *ReLU*. A função *ReLU* é dada por:

$$ReLU(z_i) = \max(0, z_i), \text{ para } i = 1, \dots, 16 \quad (17)$$

onde  $z_i$  representa o *input pesado* para a camada de escondida (hidden layer).

- $O_1$  e  $O_4$  - camada de *output* - camada formada por 4 neurónios, e em que cada neurónio tem como função de activação a função *Softmax*.

A função *Softmax*, para  $K$  classes, com  $K = 4$  (1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D), é definida por

$$S(z_k) = \frac{e^{z_k}}{\sum_{l=1}^K e^{z_l}}, \quad k = 1, \dots, K, \quad (18)$$

onde  $z_k$  representa o *input pesado* para a camada de output. Esta função possui as seguintes propriedades imediatas:

- $0 < S(z_k) < 1$
- $\sum_{k=1}^K S(z_k) = 1$

Desta forma, os 4 valores de *output* produzidos pela rede correspondem ao vector  $[pA, pB, pC, pD]$ , onde  $pS$  representa probabilidade de abrir o semáforo  $S$  (com  $S = A, B, C, D$ ). O semáforo a abrir será aquele com maior probabilidade. Em função do semáforo seleccionado pela rede, o tempo de verde é calculado usando (10).

#### 4.4 Treino das Redes Neurais

Tendo em conta os quatro conjuntos de treino construídos  $DTreino-T_{med}$ ,  $DTreino-T_{med-50\%vec1}$ ,  $DTreino-T_{med-70\%vec1}$ ,  $DTreino-T_{med-70\%mf}$ , foram treinadas quatro modelos de redes neurais, com a arquitectura apresentada no diagrama anterior.

Este tipo de modelo permite definir certas opções para treino:

- *Epoch*: uma época é quando todo o conjunto de treino passa na rede neuronal uma vez. Para o treino destas redes o número de épocas usado foi de 200;
- *BatchSize*: um conjunto de treino pode ser dividido em *batches* (subconjuntos do conjunto de treino) se não for possível passar todo o conjunto de treino pela rede neuronal de uma só vez, devido ao seu tamanho. Por defeito, o *BatchSize* foi de 12;
- *Learning Rate*: é o comprimento do passo que é usado para a actualização dos parâmetros da rede. Por defeito, este valor está definido como sendo 0.2.

As quatro redes neurais foram treinadas usando as restantes opções por defeito, com excepção das três opções supra mencionadas. Usando os conjuntos de treino  $DTreino-T_{med}$ ,  $DTreino-T_{med-50\%vec1}$ ,  $DTreino-T_{med-70\%vec1}$ ,  $DTreino-T_{med-70\%mf}$ , obtiveram-se quatro modelos de redes neurais diferentes, designadas daqui em diante por: Modelo- $T_{med}$ , Modelo- $T_{med-50\%vec1}$ , Modelo- $T_{med-70\%vec1}$ , Modelo- $T_{med-70\%mf}$ , respectivamente.

A título de exemplo, considerando o treino do Modelo- $T_{med}$  e Epoch=200, mostra-se na figura 20 como o erro (*loss function*) diminui à medida que a matriz dos pesos da rede se aproxima da matriz óptima.

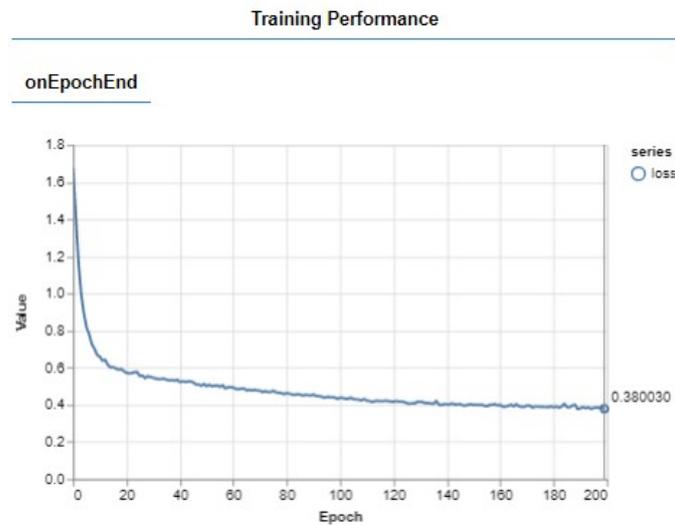


Figura 20: Desempenho da fase de treino da rede.

## 4.5 Métricas de desempenho

Para analisar o desempenho de cada um dos modelos treinados, é necessário calcular a chamada matriz de confusão. Esta matriz permite uma melhor compreensão do tipo de erros que estão a ser cometidos pelo modelo. Para construir a matriz de confusão para uma classificação multiclasse é necessário ter um conjunto de testes em que são conhecidos os respectivos valores de saída (labels)

Por forma a tornar mais fácil a compreensão da construção de uma matriz de confusão multiclasse, será primeiro introduzida a construção de uma matriz de confusão para classificação binária. Trata-se de uma matriz  $2 \times 2$ , cujas entradas são obtidas por comparação entre os valores de saída e as previsões. Os valores de saída e as previsões dadas pelo modelo são comparadas, para obter os quatro valores da matriz de confusão. Estes valores são obtidos contando o número de resultados em cada uma das seguintes categorias (ver a Tabela 3):

- *Verdadeiro - Positivo (VP)* : O valor VP representa o número de vezes em que o modelo prevê de forma correcta a classe positiva como sendo positiva;
- *Verdadeiro - Negativo (VN)* : O valor VN representa o número de vezes em que o modelo prevê correctamente a classe negativa como sendo negativa.
- *Falso - Positivo (FP)* : O valor FP representa o número de vezes em que o modelo prevê incorrectamente a classe negativa como sendo positiva.
- *Falso - Negativo (FN)* : O valor FN representa o número de previsões em que o modelo prevê incorrectamente a classe positiva como sendo negativa.

Apresenta-se na Tabela 3 a forma geral de uma matriz de confusão para classificação binária.

Tabela 3: Matriz de confusão binária.

		Classe Prevista	
		Positivo	Negativo
Classe Real	Positivo	<b>VP</b>	<b>FN</b>
	Negativo	<b>FP</b>	<b>VN</b>

Usando os quatro valores que formam a matriz de confusão, é possível calcular as três seguintes métricas de desempenho, relativo ao modelo.

A *Precision* é obtida dividindo o número de exemplos que foram classificados correctamente como positivos, pelo número de exemplos previstos como positivos:

$$Precision = \frac{VP}{VP + FP}. \quad (19)$$

A *Accuracy* é dada pela divisão entre o total de exemplos correctamente classificados, e o total de previsões feitas.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}. \quad (20)$$

A *Recall* é obtida dividindo o número de exemplos que foram classificados correctamente como positivos, pelo número de exemplos relevantes, ou seja os verdadeiros positivos mais o falsos negativos:

$$Recall = \frac{VP}{VP + FN}. \quad (21)$$

Ao contrário das matrizes de confusão binárias, nas matrizes de confusão multiclasse, não há classes positivas ou negativas.

A matriz de confusão multiclasse, em particular para quatro classes, é uma matriz de ordem 4 definida como ilustrado na Tabela 4:

Tabela 4: Matriz de confusão multiclasse

		Decisão da Rede			
		Classe 1	Classe 2	Classe 3	Classe 4
Realidade	Classe 1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$
	Classe 2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$
	Classe 3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$
	Classe 4	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$

Para construir esta matriz de confusão, as dezasseis entradas,  $x_{ij}$  com  $i, j = 1, \dots, 4$  são obtidas, comparando os valores de saída e as previsões dadas pelo modelo [29].

Os valores  $VP$ ,  $VN$ ,  $FP$  e  $FN$  para cada classe  $i$  ( $i = 1, \dots, 4$ ) serão calculados com base nas seguintes equações:

$$VP_i = \sum_{j=1}^4 x_{jj}. \quad (22)$$

$$VN_i = \sum_{\substack{j=1, \\ j \neq i}}^4 \sum_{\substack{k=1, \\ k \neq i}}^4 x_{j,k}. \quad (23)$$

$$FP_i = \sum_{\substack{j=1, \\ j \neq i}}^4 x_{j,i}. \quad (24)$$

$$FN_i = \sum_{\substack{j=1, \\ j \neq i}}^4 x_{i,j}. \quad (25)$$

Por exemplo, considerando a seguinte matriz, que ilustra a comparação entre os valores da decisão de uma rede neuronal e os valores conhecidos:

Tabela 5: Exemplo de uma matriz de confusão multi-classe de uma rede neuronal.

		Decisão da Rede			
		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Realidade	<b>1</b>	<b>58</b>	0	2	3
	<b>2</b>	5	<b>63</b>	4	4
	<b>3</b>	3	3	<b>53</b>	2
	<b>4</b>	1	2	2	<b>61</b>

Para a classe **1**, vamos calcular os valores  $VP$ ,  $VN$ ,  $FP$  e  $FN$  usando as equações (22)-(25):

- $VP_1 = 58$
- $VN_1 = 63 + 4 + 4 + 3 + 53 + 2 + 2 + 2 + 61 = 194$
- $FP_1 = 5 + 3 + 1 = 9$
- $FN_1 = 0 + 2 + 3 = 5$

Assim, as métricas de desempenho para cada classe  $i$ , com  $i = 1, \dots, 4$ , são calculadas através das seguintes fórmulas:

$$Precision_i = \frac{VP_i}{VP_i + FP_i} \quad (26)$$

$$Accuracy_i = \frac{VP_i + VN_i}{VP_i + VN_i + FP_i + FN_i}. \quad (27)$$

$$Recall_i = \frac{VP_i}{VP_i + FN_i}. \quad (28)$$

Para finalizar, outra métrica de desempenho que permite avaliar o desempenho do modelo é a *Overall Accuracy*. Esta métrica é dada pelo quociente entre o total dos verdadeiros positivos e o total de classificações ( $N$ ) realizadas pelo modelo:

$$Overall Accuracy_i = \frac{\sum_{i=1}^4 x_{ii}}{N}. \quad (29)$$

## 4.6 Validação dos Modelos

Por forma a fazer a validação dos quatro modelos gerados na fase de treino, é necessário construir quatro conjuntos de teste, tendo em consideração as respectivas fórmulas para o cálculo dos tempos médios de espera dos veículos (12), (14), (15) e (16).

Como foi referido, os conjuntos de treino foram construídos de forma manual, e, seria de esperar que a construção dos conjuntos de teste também assim fosse. Porém, este processo é bastante penoso do ponto de vista do utilizador que tem de construir os conjuntos de treino e de teste. Assim, para ultrapassar este inconveniente e tendo em consideração a experiência ganha aquando da construção manual dos conjuntos de treino, optou-se por desenvolver um algoritmo para automatizar as tarefas do polícia sinaleiro (simulador manual-teclado). Este algoritmo é designado, daqui em diante, simplesmente por *Perito*.

O *Perito* tem como *inputs* o número de veículos e o tempo médio de espera dos veículos por faixa, e dá como *output* o semáforo a activar.

O *Perito* baseia-se nos seguintes pressupostos:

- é dada prioridade à faixa com maior tempo médio de espera;
- caso existam tempos médios de espera iguais, o semáforo a activar será o da faixa que apresenta um maior número de veículos.

É de realçar a vantagem de se ter automatizado este processo. Com esta automatização não é necessário armar fisicamente os conjuntos de teste, pois, ao mesmo tempo que a simulação está a decorrer, simplesmente avalia-se e testa-se o modelo, comparando a resposta do *Perito* com a resposta do modelo.

A figura 21 ilustra o funcionamento em simultâneo da rede neuronal e do *Perito*.

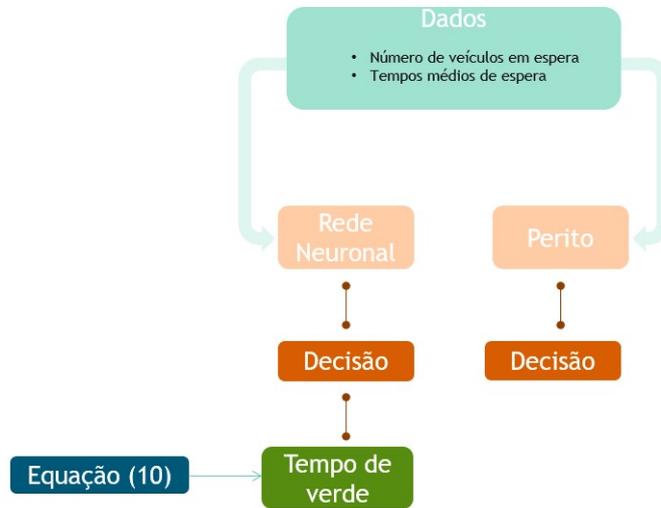


Figura 21: Modo de funcionamento do Perito e da rede neuronal, na otimização do fluxo de tráfego num cruzamento.

#### 4.6.1 Modelos- $T_{med}$ , $T_{med-50\%vec1}$ , $T_{med-70\%vec1}$ , $T_{med-70\%mf}$

Para testar cada um dos quatro modelos vão ser realizadas quatro simulações, uma por cada modelo, demorando uma hora cada uma. Para estas simulações considerou-se uma velocidade média de 12 m/s e um fluxo de 15 carros por minuto (ver figura 22).

Simulação:   Aplicar Modelo existente

Velocidade média:  m/s

Fluxo de veículos:  /min

Figura 22: Menu para a escolha do tipo de Simulação Inteligente.

Após uma hora de simulação apresentam-se de seguida as matrizes de confusão obtidas para cada modelo e as métricas de desempenho: *accuracy*, *precision* e *recall*.

**Resultados para o Modelo- $T_{med}$**  Apresenta-se a seguir a matriz de confusão para o Modelo- $T_{med}$  em que o tempo médio de espera dos veículos é dado pela equação (12).

Tabela 6: Matriz de confusão para Modelo- $T_{med}$

		Classe do Modelo			
		<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
Classe do Perito	<b>A</b>	<b>55</b>	1	1	3
	<b>B</b>	9	<b>57</b>	1	3
	<b>C</b>	1	8	<b>63</b>	6
	<b>D</b>	1	1	0	<b>59</b>

Da tabela 6 somando todas as entradas da matriz de confusão, pode concluir-se que  $N = 269$ , o que corresponde ao número total de classificações realizadas por este modelo. De todas essas classificações é possível verificar que o modelo acertou em abrir o semáforo para a faixa A 55 vezes, acertou em não abrir 198 vezes e falhou 16 vezes; acertou em abrir o semáforo para a faixa B 57 vezes e não abrir 189 vezes e falhou 23 vezes; acertou em abrir o semáforo para a faixa C 63 vezes e não abrir 189 vezes e falhou 17 vezes; acertou em abrir o semáforo para a faixa D 59 vezes e não abrir 196 vezes e falhou 14 vezes.

Os valores  $VP_i$ ,  $VN_i$ ,  $FP_i$  e  $FN_i$  para cada classe  $i = 1, \dots, 4$ , onde 1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D, são:

- $VP_1 = 55, VN_1 = 198, FP_1 = 11, FN_1 = 5$
- $VP_2 = 57, VN_2 = 189, FP_2 = 10, FN_2 = 13$
- $VP_3 = 63, VN_3 = 189, FP_3 = 2, FN_3 = 15$
- $VP_4 = 59, VN_4 = 196, FP_4 = 12, FN_4 = 2$

Conclui-se que a classe com menos classificações erradas foi a classe correspondente à faixa D.

Apresenta-se na tabela seguinte as métricas de desempenho associadas a este modelo.

Tabela 7: Tabela das métricas para Modelo- $T_{med}$ .

		Métricas		
		<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>
Classe	<b>A</b>	94.05%	0.83	0.83
	<b>B</b>	91.45 %	0.85	0.81
	<b>C</b>	93.68 %	0.97	0.81
	<b>D</b>	94.08 %	0.83	0.97

Para este modelo, obteve-se uma *Overall Accuracy* de 86.99%.

**Resultados para o Modelo- $T_{med-50\%vec1}$**  Apresenta-se a seguir a matriz de confusão para o Modelo- $T_{med-50\%vec1}$  em que o tempo médio de espera dos veículos é dado pela equação (14).

Tabela 8: Matriz de confusão para o Modelo- $T_{med-50\%vec1}$ .

		Classe do Modelo			
		A	B	C	D
Classe do Perito	A	58	1	0	5
	B	3	62	1	0
	C	1	0	63	2
	D	4	6	3	62

Da tabela 8 pode concluir-se que  $N = 271$ , o que corresponde ao número total de classificações realizadas por este modelo. De todas essas classificações é possível verificar que o modelo acertou em abrir o semáforo para a faixa A 58 vezes e acertou em não abrir 199 vezes e falhou 14 vezes; acertou em abrir o semáforo para a faixa B 62 vezes e não abrir 198 vezes e falhou 11 vezes; acertou em abrir o semáforo para a faixa C 63 vezes e não abrir 201 vezes e falhou 7 vezes; acertou em abrir o semáforo para a faixa D 62 vezes e não abrir 189 vezes e falhou 20 vezes.

Os valores  $VP_i$ ,  $VN_i$ ,  $FP_i$  e  $FN_i$  para cada classe  $i = 1, \dots, 4$ , onde 1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D, são:

- $VP_1 = 58, VN_1 = 199, FP_1 = 8, FN_1 = 6$
- $VP_2 = 62, VN_2 = 198, FP_2 = 7, FN_2 = 4$
- $VP_3 = 63, VN_3 = 201, FP_3 = 4, FN_3 = 3$
- $VP_4 = 62, VN_4 = 189, FP_4 = 7, FN_4 = 13$

Conclui-se que a classe com menos classificações erradas foi a classe correspondente à faixa C. Apresenta-se na tabela seguinte as métricas de desempenho associadas a este modelo.

Tabela 9: Tabela das métricas para o Modelo- $T_{med-50\%vec1}$ .

		Métricas		
		Accuracy	Precision	Recall
Classe	A	95.09 %	0.88	0.92
	B	98.49 %	1.0	0.94
	C	95.09 %	0.88	0.94
	D	96.23 %	0.95	0.90

Para este modelo, obteve-se uma *Overall Accuracy* de 92.45%.

**Resultados para o Modelo- $T_{med-70\%vec1}$**  Apresenta-se a seguir a matriz de confusão para o Modelo- $T_{med-70\%vec1}$  em que o tempo médio de espera dos veículos é dado pela equação (15).

Tabela 10: Matriz de confusão para o Modelo- $T_{med-70\%vec1}$ .

		Classe do Modelo			
		A	B	C	D
Classe do Perito	A	62	1	0	0
	B	1	58	0	0
	C	4	7	64	6
	D	0	2	2	67

Da tabela 10 pode concluir-se que  $N = 274$ , o que corresponde ao número total de classificações realizadas por este modelo. De todas essas classificações é possível verificar que o modelo acertou em abrir o semáforo para a faixa A 62 vezes e acertou em não abrir 206 vezes e falhou 6 vezes; acertou em abrir o semáforo para a faixa B 58 vezes e não abrir 205 vezes e falhou 11 vezes; acertou em abrir o semáforo para a faixa C 64 vezes e não abrir 191 vezes e falhou 19 vezes; acertou em abrir o semáforo para a faixa D 67 vezes e não abrir 197 vezes e falhou 10 vezes.

Os valores  $VP_i$ ,  $VN_i$ ,  $FP_i$  e  $FN_i$  para cada classe  $i = 1, \dots, 4$ , onde 1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D, são:

- $VP_1 = 62, VN_1 = 206, FP_1 = 5, FN_1 = 1$
- $VP_2 = 58, VN_2 = 205, FP_2 = 10, FN_2 = 1$
- $VP_3 = 64, VN_3 = 191, FP_3 = 2, FN_3 = 17$
- $VP_4 = 67, VN_4 = 197, FP_4 = 6, FN_4 = 4$

Conclui-se que a classe com menos classificações erradas foi a classe correspondente à faixa A. Apresenta-se na tabela seguinte as métricas de desempenho associadas a este modelo.

Tabela 11: Tabela das métricas para o Modelo- $T_{med-70\%vec1}$ .

		Métricas		
		Accuracy	Precision	Recall
Classe	A	97.81%	0.93	0.98
	B	95.99 %	0.85	0.98
	C	93.07 %	0.97	0.79
	D	96.35 %	0.92	0.94

Para este modelo, obteve-se uma *Overall Accuracy* de 87.23%.

**Resultados para o Modelo- $T_{med-70\%mf}$**  Apresenta-se a seguir a matriz de confusão para o Modelo- $T_{med-70\%mf}$  em que o tempo médio de espera dos veículos é dado pela equação (16).

Tabela 12: Matriz de confusão para o Modelo- $T_{med-70\%mf}$ .

		Classe do Modelo			
		A	B	C	D
Classe do Perito	A	65	1	2	1
	B	6	66	0	7
	C	1	5	67	6
	D	2	1	0	56

Da tabela 12 pode concluir-se que  $N = 286$ , o que corresponde ao número total de classificações realizadas por este modelo. De todas essas classificações é possível verificar que o modelo acertou em abrir o semáforo para a faixa A 65 vezes e acertou em não abrir 208 vezes e falhou 13 vezes; acertou em abrir o semáforo para a faixa B 66 vezes e não abrir 200 vezes e falhou 20 vezes; acertou em abrir o semáforo para a faixa C 67 vezes e não abrir 205 vezes e falhou 14 vezes; acertou em abrir o semáforo para a faixa D 56 vezes e não abrir 213 vezes e falhou 17 vezes.

Os valores  $VP_i$ ,  $VN_i$ ,  $FP_i$  e  $FN_i$  para cada classe  $i = 1, \dots, 4$ , onde 1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D, são:

- $VP_1 = 65, VN_1 = 208, FP_1 = 9, FN_1 = 4$
- $VP_2 = 66, VN_2 = 200, FP_2 = 7, FN_2 = 13$
- $VP_3 = 67, VN_3 = 205, FP_3 = 2, FN_3 = 12$
- $VP_4 = 56, VN_4 = 213, FP_4 = 14, FN_4 = 3$

Conclui-se que a classe com menos classificações erradas foi a classe correspondente à faixa A. Apresenta-se na tabela seguinte as métricas de desempenho associadas a este modelo.

Tabela 13: Tabela das métricas para o Modelo- $T_{med-70\%mf}$ .

		Métricas		
		Accuracy	Precision	Recall
Classe	A	95.45%	0.88	0.94
	B	93.01 %	0.90	0.84
	C	95.1 %	0.97	0.85
	D	94.06 %	0.80	0.95

Para este modelo, obteve-se uma *Overall Accuracy* de 88.81%.

Ilustra-se na figura seguinte, para uma melhor comparação dos quatro modelos, a sua *Accuracy* para cada uma das faixas bem como a *Overall Accuracy* dos modelos.

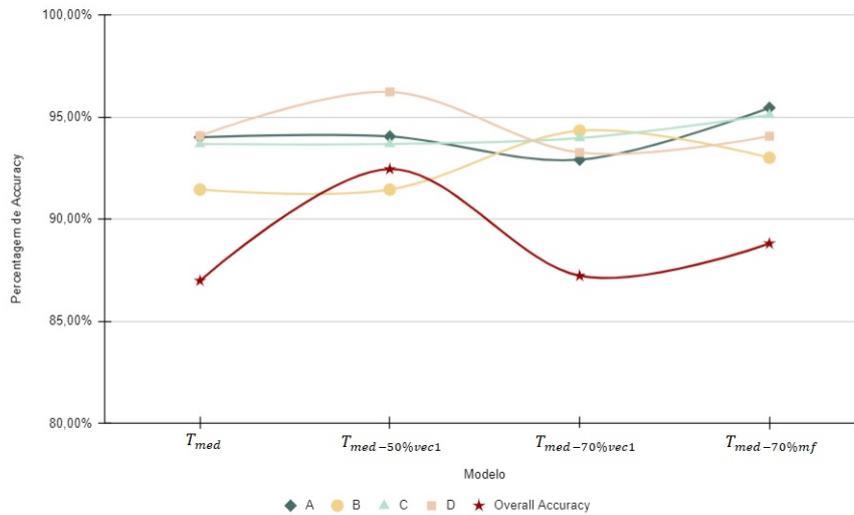


Figura 23: Variação da *Accuracy* perante os vários pesos com dados gerados de forma manual.

Na figura 23 é possível verificar que o modelo que apresenta uma maior *Overall Accuracy* é o modelo- $T_{med-50\%vec1}$  e, em contrapartida, o modelo que tem o valor mais baixo é o  $T_{med}$ .

Uma vez que estes modelos foram treinados com um conjunto de treino relativamente pequeno, decidiu-se fazer um novo treino de cada um dos modelos, com o objectivo de melhorar o desempenho dos modelos. Assim decidiu-se acrescentar aos conjuntos de treino existentes (obtidos com o simulador manual-teclado), os dados, aquando do teste/validação dos modelos, em que a decisão da rede é igual à decisão do *Perito*.

Assim, após uma hora de simulação, a cardinalidade destes conjuntos de treino passou a ser a seguinte:

- #DTreino- $T_{med}$  = 345
- #DTreino- $T_{med-50\%vec1}$  = 357
- #DTreino- $T_{med-70\%vec1}$  = 364
- #DTreino- $T_{med-70\%mf}$  = 360

A seguinte figura ilustra a evolução na construção de cada conjunto de treino.

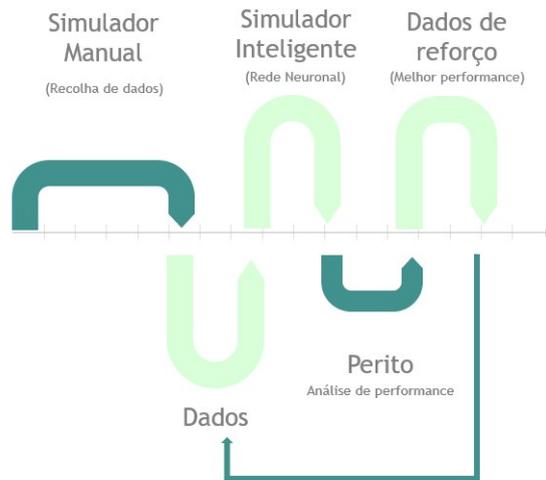


Figura 24: Esquema cronológico da construção de cada conjunto de treino.

Tendo em conta estes quatro conjuntos de treino, foram treinados os quatros modelos com a mesma configuração que é descrita na subsecção 4.4.

#### 4.6.2 Validação dos Modelos- $T_{med}$ , $T_{med-50\%vec1}$ , $T_{med-70\%vec1}$ , $T_{med-70\%mf}$ treinados com dados de reforço

**Resultados para o Modelo- $T_{med}$**  Apresenta-se a seguir a matriz de confusão para o Modelo- $T_{med}$  em que o tempo médio de espera dos veículos é dado pela equação (12).

Tabela 14: Matriz de confusão para o Modelo- $T_{med}$ .

		Classe do Modelo			
		A	B	C	D
Classe do Perito	A	58	0	2	3
	B	5	63	4	4
	C	3	3	53	2
	D	1	2	2	61

Da tabela 14 pode concluir-se que  $N = 266$ , o que corresponde ao número total de classificações realizadas por este modelo. De todas essas classificações é possível verificar que o modelo acertou em abrir o semáforo para a faixa A 58 vezes e acertou em não abrir 194 vezes e falhou 114 vezes; acertou em abrir o semáforo para a faixa B 63 vezes e não abrir 185 vezes e falhou 18 vezes; acertou em abrir o semáforo para a faixa C 53 vezes e não abrir 197 vezes e falhou 16 vezes; acertou em abrir o semáforo para a faixa D 61 vezes e não abrir 197 vezes e falhou 14 vezes.

Os valores  $VP_i$ ,  $VN_i$ ,  $FP_i$  e  $FN_i$  para cada classe  $i = 1, \dots, 4$ , onde 1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D, são:

- $VP_1 = 58, VN_1 = 194, FP_1 = 9, FN_1 = 5$
- $VP_2 = 63, VN_2 = 185, FP_2 = 5, FN_2 = 13$
- $VP_3 = 53, VN_3 = 197, FP_3 = 8, FN_3 = 8$
- $VP_4 = 61, VN_4 = 197, FP_4 = 9, FN_4 = 5$

Apresenta-se na tabela seguinte as métricas de desempenho associadas a este modelo.

Tabela 15: Tabela das métricas para o Modelo- $T_{med}$ .

		Métricas		
		Accuracy	Precision	Recall
Classe	A	94.74%	0.87	0.92
	B	93.23 %	0.93	0.83
	C	93.98 %	0.87	0.87
	D	94.74 %	0.87	0.92

Para este modelo, obteve-se uma *Overall Accuracy* de 88.35%.

**Resultados para o Modelo- $T_{med-50\%vec1}$**  Apresenta-se a seguir a matriz de confusão para o Modelo- $T_{med-50\%vec1}$  em que o tempo médio de espera dos veículos é dado pela equação (14).

Tabela 16: Matriz de confusão para o Modelo- $T_{med-50\%vec1}$ .

		Classe do Modelo			
		A	B	C	D
Classe do Perito	A	61	1	0	0
	B	1	68	0	0
	C	2	2	64	3
	D	5	2	3	61

Da tabela 16 pode concluir-se que  $N = 273$ , o que corresponde ao número total de classificações realizadas por este modelo. De todas essas classificações é possível verificar que o modelo acertou em abrir o semáforo para a faixa A 61 vezes e acertou em não abrir 203 vezes e falhou 9 vezes; acertou em abrir o semáforo para a faixa B 68 vezes e não abrir 199 vezes e falhou 6 vezes; acertou em abrir o semáforo para a faixa C 64 vezes e não abrir 199 vezes e falhou 10 vezes; acertou em abrir o semáforo para a faixa D 61 vezes e não abrir 199 vezes e falhou 13 vezes.

Os valores  $VP_i, VN_i, FP_i$  e  $FN_i$  para cada classe  $i = 1, \dots, 4$ , onde 1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D, são:

- $VP_1 = 61, VN_1 = 203, FP_1 = 8, FN_1 = 1$

- $VP_2 = 68, VN_2 = 199, FP_2 = 5, FN_2 = 1$
- $VP_3 = 64, VN_3 = 199, FP_3 = 3, FN_3 = 7$
- $VP_4 = 61, VN_4 = 199, FP_4 = 3, FN_4 = 10$

Apresenta-se na tabela seguinte as métricas de desempenho associadas a este modelo.

Tabela 17: Tabela das métricas para o Modelo- $T_{med-50\%vec1}$ .

		Métricas		
		Accuracy	Precision	Recall
Classe	A	96.7 %	0.88	0.98
	B	97.8 %	0.93	0.99
	C	96.34 %	0.96	0.90
	D	95.24 %	0.95	0.86

Para este modelo, obteve-se uma *Overall Accuracy* de 93.04%.

**Resultados para o Modelo- $T_{med-70\%vec1}$**  Apresenta-se a seguir a matriz de confusão para o Modelo- $T_{med-70\%vec1}$  em que o tempo médio de espera dos veículos é dado pela equação (15).

Tabela 18: Matriz de confusão para o Modelo- $T_{med-70\%vec1}$ .

		Classe do Modelo			
		A	B	C	D
Classe do Perito	A	64	5	2	2
	B	0	57	9	0
	C	1	0	68	0
	D	10	2	5	57

Da tabela 18 pode concluir-se que  $N = 282$ , o que corresponde ao número total de classificações realizadas por este modelo. De todas essas classificações é possível verificar que o modelo acertou em abrir o semáforo para a faixa A 64 vezes e acertou em não abrir 198 vezes e falhou 20 vezes; acertou em abrir o semáforo para a faixa B 57 vezes e não abrir 199 vezes e falhou 16 vezes; acertou em abrir o semáforo para a faixa C 68 vezes e não abrir 199 vezes e falhou 17 vezes; acertou em abrir o semáforo para a faixa D 57 vezes e não abrir 199 vezes e falhou 19 vezes.

Os valores  $VP_i, VN_i, FP_i$  e  $FN_i$  para cada classe  $i = 1, \dots, 4$ , onde 1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D, são:

- $VP_1 = 64, VN_1 = 198, FP_1 = 11, FN_1 = 9$
- $VP_2 = 57, VN_2 = 199, FP_2 = 7, FN_2 = 9$
- $VP_3 = 68, VN_3 = 199, FP_3 = 16, FN_3 = 1$

- $VP_4 = 57, VN_4 = 199, FP_4 = 2, FN_4 = 17$

Apresenta-se na tabela seguinte as métricas de desempenho associadas a este modelo.

Tabela 19: Tabela das métricas para o Modelo- $T_{med-70\%vec1}$ .

		Métricas		
		Accuracy	Precision	Recall
Classe	A	92.91%	0.85	0.88
	B	94.33 %	0.89	0.86
	C	93.97 %	0.81	0.99
	D	93.26 %	0.97	0.77

Para este modelo, obteve-se uma *Overall Accuracy* de 91.61%.

**Resultados para o Modelo- $T_{med-70\%mf}$**  Apresenta-se a seguir a matriz de confusão para o Modelo- $T_{med-70\%mf}$  em que o tempo médio de espera dos veículos é dado pela equação (16).

Tabela 20: Matriz de confusão para o Modelo- $T_{med-70\%mf}$ .

		Classe do Modelo			
		A	B	C	D
Classe do Perito	A	52	0	1	0
	B	11	59	3	3
	C	1	3	65	0
	D	4	4	0	65

Da tabela (20) pode concluir-se que  $N = 271$ , o que corresponde ao número total de classificações realizadas por este modelo. De todas essas classificações é possível verificar que o modelo acertou em abrir o semáforo para a faixa A 52 vezes e acertou em não abrir 202 vezes e falhou 17 vezes; acertou em abrir o semáforo para a faixa B 59 vezes e não abrir 199 vezes e falhou 24 vezes; acertou em abrir o semáforo para a faixa C 65 vezes e não abrir 199 vezes e falhou 8 vezes; acertou em abrir o semáforo para a faixa D 65 vezes e não abrir 199 vezes e falhou 11 vezes.

Os valores  $VP_i, VN_i, FP_i$  e  $FN_i$  para cada classe  $i = 1, \dots, 4$ , onde 1-Semáforo A, 2-Semáforo B, 3-Semáforo C e 4-Semáforo D, são:

- $VP_1 = 52, VN_1 = 202, FP_1 = 16, FN_1 = 1$
- $VP_2 = 59, VN_2 = 199, FP_2 = 7, FN_2 = 17$
- $VP_3 = 65, VN_3 = 199, FP_3 = 4, FN_3 = 4$
- $VP_4 = 65, VN_4 = 199, FP_4 = 3, FN_4 = 8$

Apresenta-se na tabela seguinte as métricas de desempenho associadas a este modelo.

Tabela 21: Tabela das métricas para o Modelo- $T_{med-70\%mf}$ .

		Métricas		
		Accuracy	Precision	Recall
Classe	A	93.73%	0.76	0.98
	B	91.14 %	0.89	0.78
	C	97.05 %	0.94	0.94
	D	95.94 %	0.96	0.89

Para este modelo, obteve-se uma *Overall Accuracy* de 88.93%.

Ilustra-se na figura seguinte, para uma melhor comparação dos quatro modelos, a sua *Accuracy* para cada uma das faixas bem como a *Overall Accuracy* dos modelos.

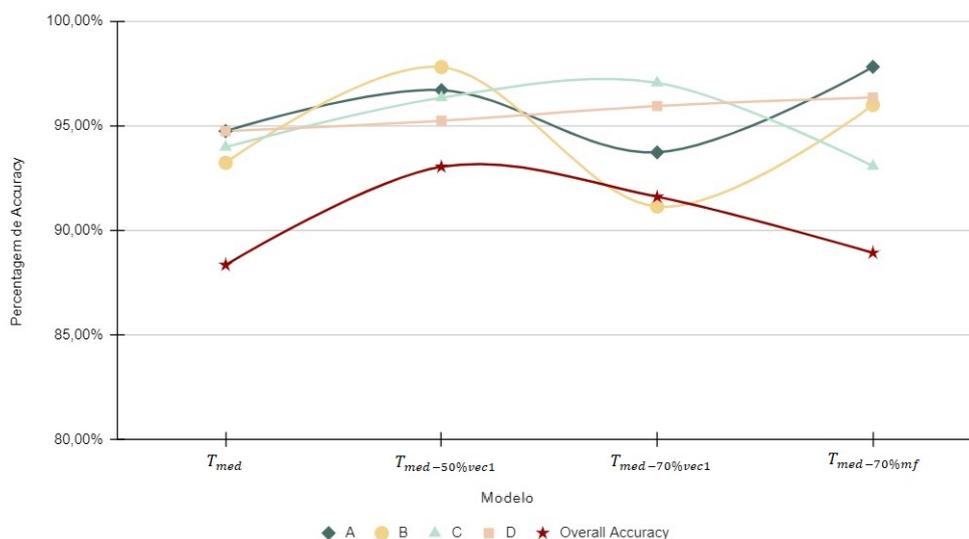


Figura 25: Variação da *Accuracy* e *Overall Accuracy* perante os vários pesos com dados de reforço.

Na figura 25 é possível verificar que o modelo que apresenta uma maior *Overall Accuracy* é o modelo- $T_{med-50\%vec1}$  e, em contrapartida, o modelo que tem o valor mais baixo é o  $T_{med}$ .

#### 4.6.3 Comparação dos Modelos treinados sem e com dados de reforço

Apresentamos nesta secção um estudo comparativo do desempenho dos Modelos- $T_{med}$ ,  $T_{med-50\%vec1}$ ,  $T_{med-70\%vec1}$ ,  $T_{med-70\%mf}$  treinados sem e com dados de reforço

Para cada um dos modelos, apresenta-se nas tabelas seguintes o tempo médio de espera por faixa (calculado pela fórmula (12)), o número de veículos despachados e a *Overall Accuracy*:

Tabela 22: Resultados da simulação com os modelos- $T_{med}$ ,  $T_{med-50\%vec1}$ ,  $T_{med-70\%vec1}$ ,  $T_{med-70\%mf}$  sem dados de reforço.

Modelos	$T_{medA}$	$T_{medB}$	$T_{medC}$	$T_{medD}$	$N_{vd}$	Overall Accuracy
Modelo- $T_{med}$	35 s	36 s	37 s	36 s	899	86.99 %
Modelo- $T_{med-50\%vec1}$	35 s	38 s	36 s	37 s	914	92.45 %
Modelo- $T_{med-70\%vec1}$	35 s	34 s	36 s	31 s	917	87.23 %
Modelo- $T_{med-70\%mf}$	33 s	36 s	36 s	34 s	943	88.81 %

Tabela 23: Resultados da simulação com os modelos- $T_{med}$ ,  $T_{med-50\%vec1}$ ,  $T_{med-70\%vec1}$ ,  $T_{med-70\%mf}$  com dados de reforço.

Modelos	$T_{medA}$	$T_{medB}$	$T_{medC}$	$T_{medD}$	$N_{vd}$	Overall Accuracy
Modelo- $T_{med}$	38 s	39 s	39 s	38 s	893	88.35 %
Modelo- $T_{med-50\%vec1}$	35 s	35 s	36 s	38 s	906	93.04 %
Modelo- $T_{med-70\%vec1}$	35 s	34 s	37 s	34 s	895	91.61 %
Modelo- $T_{med-70\%mf}$	32 s	35 s	36 s	37 s	900	88.93 %

Das tabelas 22 e 23 conclui-se que, em geral, o tempo médio de espera é inferior para os modelos treinados com os dados manuais, e o número de veículos despachados é superior. Porém, a *Overall Accuracy* é superior para a simulação feita com os modelos treinados com dados de reforço. Assim, pode-se concluir que, como esperado, o aumento da cardinalidade dos conjuntos de treino foi benéfico para a rede, resultando em decisões com maior precisão.

No que diz respeito ao desempenho dos diferentes modelos, pode concluir-se que os modelos- $T_{med-50\%vec1}$  e  $T_{med-70\%mf}$  são os modelos que apresentam um maior número de veículos despachados. O pior resultado foi obtido para o Modelo- $T_{med}$ , o qual foi treinado com dados que atribuem a mesma importância a todos os veículos.

#### 4.7 Uso do desvio-padrão do tempo médio de espera como input

Nesta secção, procurou-se averiguar se o desvio-padrão do tempo médio de espera poderia ter influência no desempenho dos modelos. Para fazer este estudo, optou-se por treinar o Modelo- $T_{med}$ , visto que este modelo apresentou uma menor *Overall Accuracy* nas simulações anteriores.

A expressão matemática que permite calcular esta medida de dispersão é dada por [30]:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (T_i - T_{med})^2}{n}}, \quad (30)$$

onde  $T_{med}$  corresponde à média do tempo de espera dos veículos  $x_i$  com  $i = 1, \dots, n$ , e  $T_i$  representa o tempo de espera do veículo  $x_i$ , de uma determinada faixa.

Assim, definiu-se uma rede neuronal com 12 *inputs*, a qual foi treinada com o novo conjunto de treino DTreino- $T_{med-\sigma}$ , obtido de forma manual durante 30 minutos usando o simulador manual-teclado. A figura 26 ilustra um

elemento do conjunto de treino  $D_{Treino-T_{med-\sigma}}$  (ficheiro *json*). Após a construção deste conjunto de treino, a sua cardinalidade foi de  $\#D_{Treino-T_{med-\sigma}} = 118$ .

É de salientar que os atributos são: o número de carros, o tempo médio de espera e o desvio-padrão, para cada faixa (primeiras 12 linhas da Figura 26), e a label (rótulo) é o semáforo a activar.

Portanto, tendo em consideração o conjunto de treino  $D_{Treino-T_{med-\sigma}}$ , foi treinada esta nova rede neuronal com a configuração que é descrita na subsecção 4.4. Este novo modelo é designado por Modelo- $T_{med-\sigma}$ .

```
{
  "numeroCarrosA": 4,
  "numeroCarrosB": 0,
  "numeroCarrosC": 0,
  "numeroCarrosD": 1,
  "TempoEsperaA": 26,
  "TempoEsperaB": 0,
  "TempoEsperaC": 0,
  "TempoEsperaD": 20,
  "DesvioPaA": 13,
  "DesvioPaB": 0,
  "DesvioPaC": 0,
  "DesvioPaD": 0,
  "Semaforo": "A",
  "TempoVerdeManual": 14
},
```

Figura 26: Parte do Dataset para treino do Modelo- $T_{med-\sigma}$ .

Após o treino do Modelo- $T_{med-\sigma}$ , procedeu-se ao seu teste/validação. Para este teste, foi efectuada uma simulação de uma hora em que se considerou uma velocidade média de 12 m/s e um fluxo de 15 carros por minuto.

Apresentam-se na tabela 24 os resultados obtidos para o Modelo- $T_{med-\sigma}$  após a simulação de uma hora realizada. Nesta tabela apresenta-se para cada faixa, o tempo médio de espera ( $T_{med}$ ), a mediana dos desvios-padrão ( $\sigma_{mediana}$ ) bem como o seu valor máximo ( $\sigma_{max}$ ), o seu valor mínimo ( $\sigma_{min}$ ) e a sua média ( $\sigma_{med}$ ).

A mediana dos desvios-padrão, para cada faixa, foi calculada usando a seguinte fórmula [40]:

$$\sigma_{mediana} = \begin{cases} \sigma(\lfloor \frac{n+1}{2} \rfloor), & \text{se } n \text{ for ímpar} \\ \frac{\sigma(\lfloor \frac{n}{2} \rfloor) + \sigma(\lfloor \frac{n}{2} \rfloor + 1)}{2}, & \text{se } n \text{ for par} \end{cases} \quad (31)$$

onde  $n$  corresponde à posição do valor do desvio-padrão numa lista que contém todos os desvios-padrão (obtidos para cada ciclo), ordenados por faixa.

Tabela 24: Resultados da simulação com o modelo- $T_{med-\sigma}$

Faixa	$T_{med}$	$\sigma_{mediana}$	$\sigma_{min}$	$\sigma_{max}$	$\sigma_{med}$	$N_{vd}$
<b>A</b>	36 s	4 s	0 s	19 s	2 s	232
<b>B</b>	35 s	4 s	0 s	21 s	2 s	231
<b>C</b>	35 s	4 s	0 s	19 s	2 s	224
<b>D</b>	37 s	4 s	0 s	24 s	2 s	202

Um análise da tabela 24 permite concluir que os resultados relativos ao tempo médio de espera global melhoraram, havendo um decréscimo de 3 segundos em comparação com os resultados obtidos para o Modelo- $T_{med}$  - tabela 23. Porém, no que diz respeito ao número de veículos despachados, verificou-se uma diminuição ligeira, tendo sido despachados menos 4 veículos. Isto pode dever-se à dificuldade de se ter em conta o desvio-padrão, aquando da geração dados de treino de forma manual para o conjunto  $DTreino-T_{med-\sigma}$ . Os valores de  $\sigma_{mediana}$  e  $\sigma_{med}$  foram de 4 s e 2 s, respectivamente, para todas as faixas. O desvio-padrão máximo foi de 24 s, sendo obtido para a faixa D.

Dada a dificuldade de construir o conjunto de treino  $DTreino-T_{med-\sigma}$  de forma manual, devido à dificuldade em dar-se a devida atenção ao desvio-padrão, sem descuidar as variáveis tempo médio de espera e o número de veículos por faixa, podemos concluir que os resultados obtidos com o Modelo- $T_{med-\sigma}$  são muito semelhantes aos obtidos com o Modelo- $T_{med}$ .

## 5 Simulador Convencional versus Simulador Inteligente

No capítulo anterior verificou-se que o simulador inteligente permitiu obter melhores resultados quando comparado com o Simulador Convencional. Verificou-se também que o Modelo- $T_{med-50\%vec1}$  foi aquele que permitiu obter um valor mais baixo de tempos médios de espera e um valor mais elevado de veículos despachados.

Assim, neste capítulo apresenta-se um estudo comparativo entre os resultados obtidos com os semáforos convencionais e os semáforos inteligentes com os dois modelos seguintes:

- Modelo- $T_{med-50\%vec1}$  (com dados de reforço) que calcula os tempos médios de espera através da equação (14), uma vez que este foi o modelo que obteve melhores resultados comparativamente com os restantes que não têm em consideração o desvio-padrão. Os resultados com este modelo apresentaram um ligeiro aumento do número de veículos despachados bem como uma *Overall Accuracy* um pouco mais elevada.
- Modelo- $T_{med-\sigma}$  que tem em consideração o desvio-padrão e que usa a equação (12) para o cálculo dos tempos médios de espera.

A figura 27 ilustra a evolução do trabalho desenvolvido nesta dissertação, desde a implementação de raiz do simulador convencional, até ao desenvolvimento do simulador inteligente.

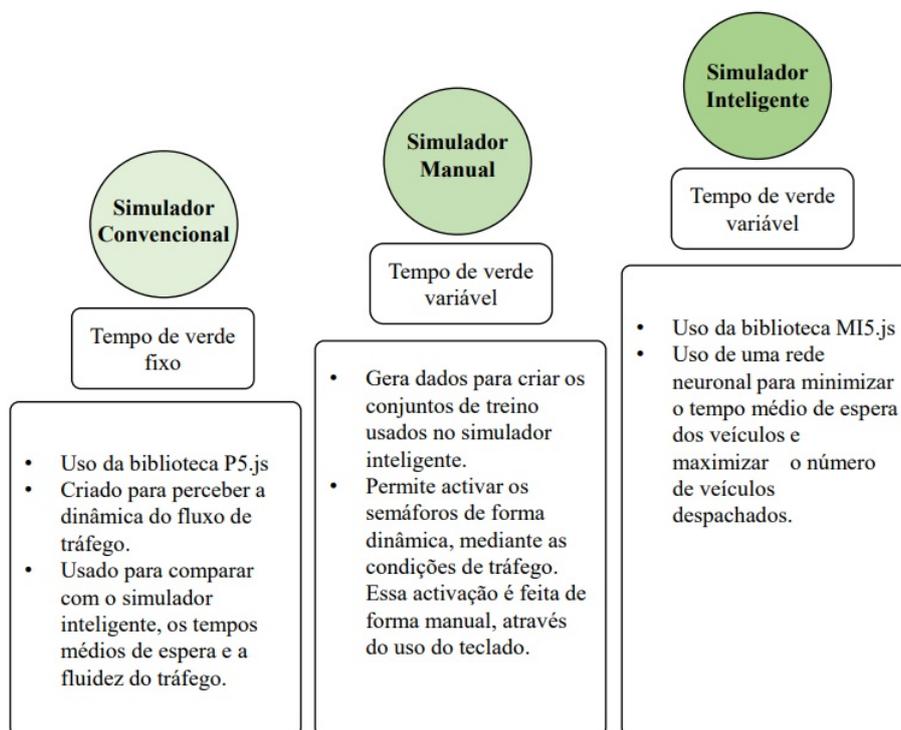


Figura 27: Evolução do trabalho desenvolvido nesta dissertação, desde a implementação de raiz do simulador convencional, até ao desenvolvimento do simulador inteligente.

A figura 28 mostra os tempos médios de espera para cada faixa, obtidos pelo simulador convencional, e pelos Modelo- $T_{med-50\%vec1}$  e Modelo- $T_{med-\sigma}$ .

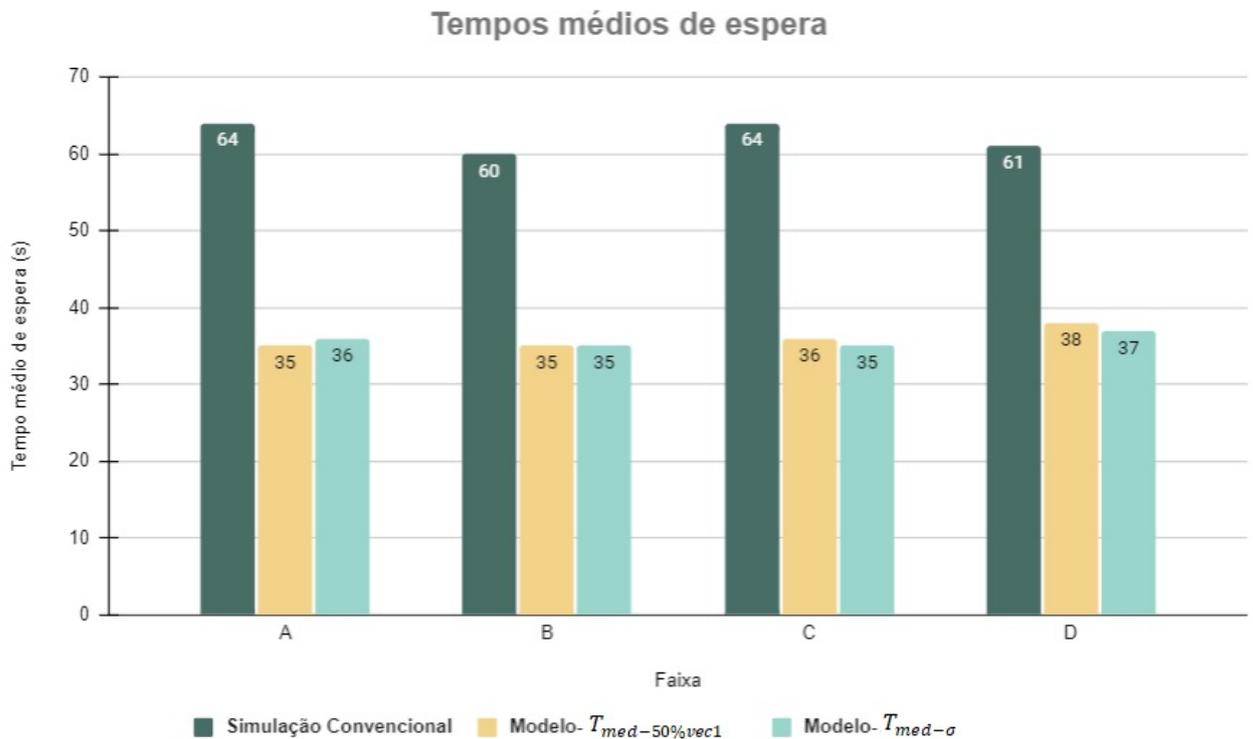


Figura 28: Tempos médios de espera obtidos com os Modelos- $T_{med-50\%vec1}$ ,  $T_{med-\sigma}$  e o simulador convencional, para as quatro faixas do cruzamento.

Como se pode observar, para condições de tráfego semelhantes, a simulação convencional apresenta um maior tempo médio de espera, enquanto que as simulações inteligentes conseguem minimizar esse tempo de espera. A presença do desvio-padrão nos dados de treino do modelo resulta em tempos de espera ligeiramente inferiores.

Relativamente ao número de carros despachados, vemos que há um ligeiro aumento de  $N_{vd}$  nas simulações inteligentes (ver tabelas 23 e 24).

Na figura 29 mostra-se o número médio de paragens sofridas por veículo. É possível concluir que este número é relativamente mais elevado nas simulações com semáforos inteligentes. Isto acontece porque os Modelos- $T_{med-50\%vec1}$  e  $T_{med-\sigma}$  vão alterando os tempos de verde ao longo da simulação, e simultaneamente, estão sempre a chegar mais carros a cada uma das faixas. Na simulação convencional o tempo de verde é fixo, e o tempo de 30 s é suficiente para despachar todos os veículos de cada faixa.

O uso de um fluxo de carros mais elevado iria resultar num maior número de paragens, mesmo para o simulador convencional.

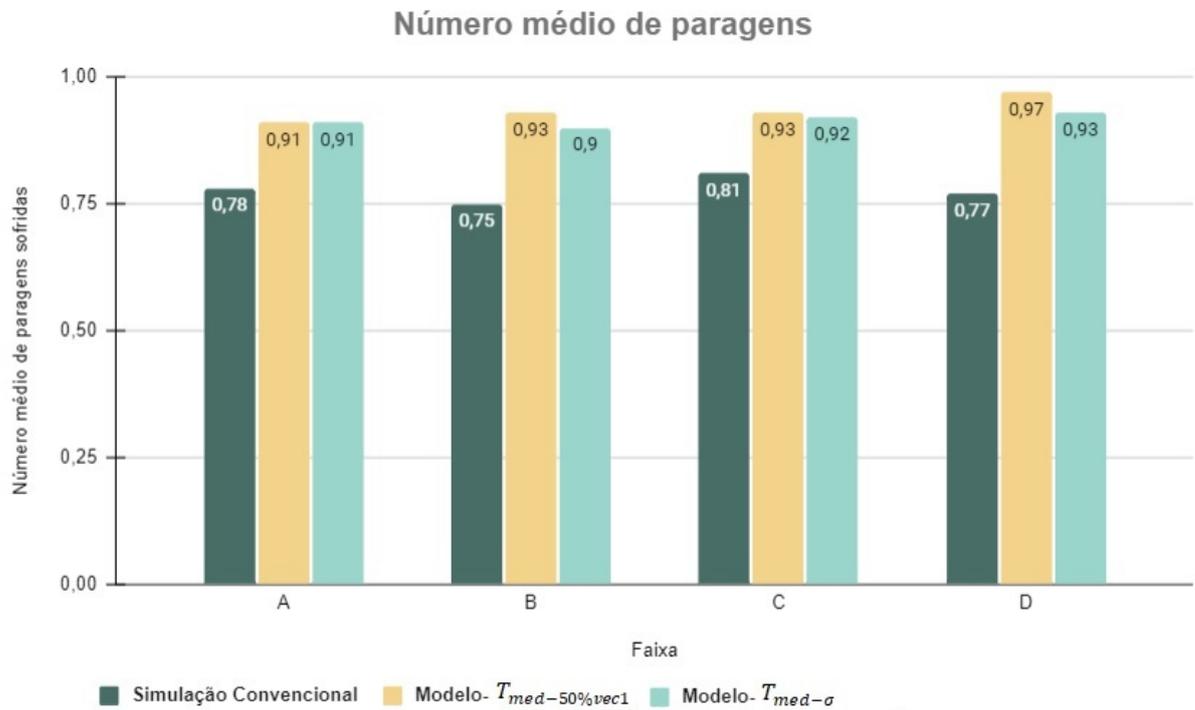


Figura 29: Número médio de paragens obtido com os Modelos- $T_{med-50\%vec1}$ ,  $T_{med-\sigma}$  e o simulador convencional, para as quatro faixas do cruzamento.

Para concluir, pode dizer-se que os modelos inteligentes revelam um desempenho muito superior ao do modelo convencional, e que, o uso do desvio-padrão como *input* não resulta em grande diferença entre o Modelo- $T_{med-50\%vec1}$  e o Modelo- $T_{med-\sigma}$ .

## 6 Conclusões e Trabalho Futuro

### 6.1 Conclusões

Nesta dissertação foram desenvolvidas novas ferramentas para a optimização do tráfego num cruzamento. Inicialmente foi desenvolvido um simulador, designado por simulador convencional, que permite modelar o comportamento de cada veículo durante todo o tempo de simulação. Para o desenvolvimento deste simulador usou-se a biblioteca *p5.js* [38] *open-source* escrita na linguagem *JavaScript*, e programação em HTML e CSS. Com o auxílio destas ferramentas de programação, foi possível desenvolver uma interface gráfica dinâmica, que representa o cruzamento bem como o fluxo de tráfego, em estudo. Este simulador foi desenvolvido tendo por base os seguintes pressupostos: um cruzamento tem quatro faixas de rodagem; quando o semáforo está verde para uma faixa, os veículos dessa faixa podem seguir em qualquer direcção; cada semáforo tem apenas duas cores (verde/vermelho), cada uma com um tempo fixo de 30 segundos; para cada faixa existe um processo aleatório para gerar os veículos. Por defeito, por forma a gerar aleatoriedade, a cada 4 segundos é colocado um veículo numa das faixas.

Foi ainda desenvolvida de forma empírica uma nova função, que quantifica o tempo que um veículo, numa determinada posição  $P_{vei}$ , demora a transpor o cruzamento.

Com o objectivo de avaliar a dinâmica do simulador convencional, foi realizada uma simulação com duração de 1 hora, com uma velocidade média de 12 m/s, tempo de verde fixo de 30 s e um fluxo de 15 carros por minuto. Concluiu-se que o tempo médio de espera dos veículos,  $T_{med}$ , varia entre 60 s e 64 s para as quatro faixas. O tempo máximo de espera,  $T_{max}$ , é diferente para cada uma das faixas. O tempo de verde fixo (30 s) não foi suficiente para escoar todos os veículos que se encontravam em espera, fazendo o tempo médio de espera,  $T_{med}$ , em cada faixa e, consequentemente, o correspondente tempo máximo de espera,  $T_{max}$ .

Relativamente ao número médio de paragens em cada faixa, verificou-se que houve um ligeiro aumento na faixa C e consequentemente um maior tempo médio de espera nessa mesma faixa. Relativamente ao número total de veículos despachados durante o tempo de verde,  $N_{vd}$ , este foi ligeiramente superior na faixa D. Estes valores, número médio de paragens, tempo médio de espera e número de veículos despachados, variam de faixa para faixa, uma vez que os factores aleatórios (como por exemplo a injeção de carros) influenciam as condições de tráfego.

Tendo em conta as limitações do simulador convencional, foi então desenvolvido um simulador inteligente, baseado em modelos de inteligência artificial (mais precisamente, redes neuronais). Para o desenvolvimento desses modelos inteligentes, foi preciso treinar a rede neuronal com determinados conjuntos de dados, ditos conjuntos de treino. Para obter os conjuntos de dados de treino foi necessário desenvolver um simulador manual (que tem por base o simulador convencional) usando um método que permite activar o semáforo manualmente através do teclado do computador. O utilizador actua como se fosse um polícia sinaleiro a olhar para as quatro faixas e a tomar a decisão de qual o semáforo a activar, tendo em conta o número de veículos e o tempo médio de espera, em cada faixa. Foram criados quatro conjuntos de treino, mediante o peso atribuído às diferentes partes da fila: DTreino- $T_{med}$  - atribuição do mesmo peso a todos os veículos da faixa; DTreino- $T_{med-50\%vec1}$  - atribuição de um peso de 50% ao primeiro veículo da fila e 50% aos restantes veículos dessa fila; DTreino- $T_{med-70\%vec1}$  - atribuição de um peso de 70% ao primeiro veículo da fila e 30% aos restantes veículos dessa fila; DTreino- $T_{med-70\%mf}$  - atribuição de um peso de 70% à primeira metade da fila e 30% à outra metade. Estes conjuntos de treino deram origem a quatro modelos de redes neuronais diferentes, designadas por: Modelo- $T_{med}$ , Modelo- $T_{med-50\%vec1}$ , Modelo- $T_{med-70\%vec1}$ , Modelo- $T_{med-70\%mf}$ .

Por forma a fazer a validação dos quatro modelos gerados pela rede, foi necessário construir quatro conjuntos de teste. Os conjuntos de treino foram construídos de forma manual, e seria de esperar que a construção dos conjuntos

de teste também assim fosse. Porém, este processo é bastante penoso do ponto de vista do utilizador que tem de construir os conjuntos de treino e de teste. Assim, para ultrapassar este inconveniente e tendo em consideração a experiência ganha aquando da construção manual dos conjuntos de treino, optou-se por desenvolver um algoritmo para automatizar as tarefas do polícia sinaleiro (simulador manual-teclado). Este algoritmo é designado por *Perito*.

O *Perito* tem, como *inputs*, o número de veículos e o tempo médio de espera dos veículos por faixa, e, dá como *output* o semáforo a activar. Este baseia-se nos seguintes pressupostos: é dada prioridade à faixa com maior tempo médio de espera; caso existam tempos médios de espera iguais, o semáforo a activar será o da faixa que apresenta um maior número de veículos. É de realçar a vantagem de se ter automatizado este processo. Com esta automatização não é necessário armazenar fisicamente o conjunto de teste, pois, ao mesmo tempo que a simulação está a decorrer, pode-se avaliar e testar o modelo, comparando as respostas do *Perito* com as respostas do modelo. Para avaliar o desempenho dos modelos, foram usadas métricas de desempenho como a Accuracy, a Precision, a Recall e a Overall Accuracy. Concluiu-se que o modelo com melhor desempenho foi o Modelo- $T_{med-50\%vec1}$ .

Uma vez que estes modelos foram treinados com um conjunto de treino pequeno, com o objectivo de melhorar o desempenho destes modelos, decidiu-se voltar a fazer o treino de cada um destes modelos. Assim decidiu-se acrescentar aos conjuntos de treino existentes (obtidos com o simulador manual-teclado), os dados, aquando do teste e validação dos modelos, em que a decisão da rede é igual à decisão do *Perito*. Concluiu-se que a Overall Accuracy melhorou para todos os modelos treinados com estes conjuntos de treino reforçados.

Deve ser realçado o facto de que o *Perito* não é um substituto da rede neuronal. O *Perito* é baseado num conjunto de premissas que não se alteram mediante o estado do tráfego, enquanto que a rede neuronal pode ir aprendendo ao longo tempo, estando mais apta a reagir em situações adversas, modelando-se a cada evento que surja no cruzamento.

Neste trabalho, procurou-se ainda averiguar se o desvio-padrão do tempo médio de espera poderia ter influência no desempenho dos modelos. Para fazer este estudo, foi criado manualmente um conjunto de dados de treino (designado por DTreino- $T_{med-\sigma}$ ), que, para além de ter em consideração o número de carros por faixa e o tempo médio de espera (dado pela equação (12)) usa a informação sobre o desvio-padrão, para tomar a decisão de abrir ou fechar um determinado semáforo.

Dado que o modelo Modelo- $T_{med}$  apresentou uma menor Overall Accuracy (quando comparado com os restantes modelos treinados), foi treinado um novo modelo com os dados de treino DTreino- $T_{med-\sigma}$ , designado por Modelo- $T_{med-\sigma}$ .

Tendo em conta a dificuldade em treinar o modelo manualmente, dando a devida atenção ao desvio-padrão, sem descurar as variáveis tempo médio de espera e o número de veículos por faixa, concluiu-se que os resultados obtidos com o desvio-padrão como *input*, são muito semelhantes aos obtidos com os modelos sem desvio-padrão com *input*.

Por fim, concluiu-se que o simulador inteligente tem uma performance muito melhor do que a do simulador de tempos fixos, resultando numa diminuição de aproximadamente 50% dos tempos médios de espera dos veículos na fila, o que prova que a abordagem inteligente consegue otimizar os tempos médios de espera dos veículos num cruzamento.

De notar que a implementação de um simulador de tráfego não é trivial, sendo um processo bastante complexo e trabalhoso, uma vez que é necessário fazer pesquisas bastante abrangentes para se conhecer em concreto toda a dinâmica das condições de tráfego.

Deve ainda ser realçado que por muita tecnologia que exista na optimização dos semáforos, temos sempre que olhar para o urbanismo de uma cidade como um todo, desde a arquitectura da via onde queremos implementar

semáforos inteligentes até ao tipo de veículos nela circulam.

## 6.2 Trabalho Futuro

Tendo em conta o trabalho desenvolvido nesta dissertação, são descritos em seguida, possíveis desenvolvimentos futuros:

- Realizar mais simulações com parâmetros de *input* mais diversificados, ou seja, usar diferentes valores de velocidade média e de fluxo de carros (nesta dissertação usou-se uma velocidade média de 12 m/s e um fluxo de 15 carros por minuto), permitindo assim obter-se uma visão mais alargada do funcionamento e desempenho dos modelos.
- O estudo do efeito de diferentes fluxos de veículos nas diferentes faixas.
- Desenvolvimento de uma equação análoga à equação (10), envolvendo mais parâmetros reais e permitindo, assim, uma descrição mais completa do tempo que um determinado veículo demora a transpor o cruzamento.
- Desenvolver uma rede neuronal que, para além de tomar a decisão de qual o semáforo a abrir, também conseguisse aprender e posteriormente prever, qual o tempo de verde a atribuir a cada semáforo.
- Introduzir um novo parâmetro no simulador para limitar à partida qual o número máximo de veículos a atravessar o cruzamento, por faixa, quando o semáforo for activado e existam veículos nas outras faixas.
- Transpor este estudo para um cruzamento real. Para isso, seria necessário realizar um estudo sobre o comportamento de tráfego num dado cruzamento de uma cidade, por exemplo, a cidade de Braga, de modo a poder recolher dados de tráfego reais e consequentemente treinar os modelos aqui propostos com os dados desse cruzamento. Na figura 30 mostra-se uma sugestão realizada por Mikael Colville-Andersen [8] sobre como melhorar um cruzamento na cidade de Braga. Este seria um bom cruzamento para implementação dos semáforos inteligentes aqui propostos nesta dissertação.



Figura 30: Exemplo de uma intersecção da cidade de Braga [8].

- Generalizar, para um conjunto de cruzamentos de uma cidade, o caso de estudo apresentado nesta dissertação para um cruzamento.
- Por fim, otimizar a rede de cruzamentos inteligentes, tendo em conta o tipo de veículos (por exemplo, veículos de transporte colectivo de passageiros e veículos de emergência), passagens de peões, a poluição, o consumo de combustível, etc.

## Referências

- [1] S Farzad Ahmadi, Austin S Berrier, William M Doty, Pat G Greer, Mohammad Habibi, Hunter A Morgan, Josam HC Waterman, Nicole Abaid, and Jonathan B Boreyko. Latent heat of traffic moving from rest. *New Journal of Physics*, 19(11):113034, 2017.
- [2] Khalid Al-Khateeb and Jaiz AY Johari. Intelligent dynamic traffic light sequence using rfid. pages 1367–1372, 2008.
- [3] Oruc Altintasi, Hediye Tuydes-Yaman, and Kagan Tuncay. Detection of urban traffic patterns from floating car data (fcd). *Transportation research procedia*, 22:382–391, 2017.
- [4] Sylvain Lassarre et Michel Roussignol Antoine Tordeux. A study of the emergence of kinematic waves in targeted state car-following models of traffic. <https://journals.openedition.org/cybergeo/23660>, mai 2011. Accessed on 2020-09-03.
- [5] Nurul Nasuha Nor Azlan and Munzilah Md Rohani. Overview of application of traffic simulation model. 150:03006, 2018.
- [6] Tom Bellemans, Bart De Schutter, and Bart De Moor. Models for traffic control. *JOURNAL A*, 43(3/4):13–22, 2002.
- [7] João Miguel Branco de Brito. Caracterização da flutuação do tráfego na cidade de lisboa. 2012.
- [8] Mikael Colville-Andersen. 5 minute urbanism - braga, portugal - with mikael colville-andersen. <https://www.youtube.com/watch?v=fs1WEcyrOBg&t=632s>, Ago 2020. Accessed on 2020-08-31.
- [9] Sérgio Costa. C-its pilot in the lisbon urban node. <https://docplayer.com.br/75212413-C-its-pilot-in-the-lisbon-urban-node.html>, Nov 2017. Accessed on 2020-08-31.
- [10] Teresa M. Costa. Psp: Há 135 anos a zelar pela segurança pública. <https://correiodominho.pt/noticias/psp-ha-135-anos-a-zelar-pela-seguranca-publica/61798>, Ago 2020. Accessed on 2020-09-02.
- [11] Secretaria Geral da Administração Interna. Ecall. <https://www.sg.mai.gov.pt/Tecnologias/112pt/ECALL/Paginas/default.aspx>, Jun 2018. Accessed on 2020-09-03.
- [12] Gestão de Frotas da Cartrack. Como funcionam os sistemas de localização de viaturas por gps? <https://www.cartrack.pt/localizacao-gps-viaturas/>, Dez 2017. Accessed on 2020-09-03.
- [13] Bart De Schutter and Bart De Moor. Optimal traffic light control for a single intersection. *European Journal of Control*, 4(3):260–276, 1998.
- [14] Vital Tech Soluções Eletrônicas. Sat – sistema de análise de tráfego vital tech. <https://vitaltechpaineis.wordpress.com/tag/laco-indutivo/>, Out 2015. Accessed on 2020-09-03.
- [15] Carrillo-González José Gerardo, Arámburo-Lizárraga Jesús, and Ortega-Magaña Ricardo. Modeling the turning speed and car following behaviors of autonomous vehicles in a virtual world. *Ingeniería, Investigación y Tecnología*, 16(3):391–405, 2015.

- [16] Bilal Ghazal, Khaled ElKhatib, Khaled Chahine, and Mohamad Kherfan. Smart traffic light control system. pages 140–145, 2016.
- [17] Nuno Gregório, Ana Bastos Silva, and Alvaro Seco. Limites de velocidade variáveis–definição do domínio e de critérios de aplicação.
- [18] Zawar H Khan, Waheed Imran, Sajid Azeem, Khurram S Khattak, T Aaron Gulliver, and Muhammad Sagheer Aslam. A macroscopic traffic model based on driver reaction and traffic stimuli. *Applied Sciences*, 9(14):2848, 2019.
- [19] Khaled Hamad and Hassan Abuhamda. Estimating base saturation flow rate for selected signalized intersections in doha, qatar. *Journal of Traffic and Logistics Engineering Vol*, 3(2), 2015.
- [20] Marci Houghlen. History of the traffic light. <https://www.motorbiscuit.com/history-of-the-traffic-light/>, Sept 2019. Accessed on 2020-09-03.
- [21] (Institute of Urban Traffic Research). ITS Development by Japanese Police. 1996.
- [22] Cihan Karakuzu and Osman Demirci. Fuzzy logic based smart traffic light simulator design and hardware implementation. *Applied Soft Computing*, 10(1):66–73, 2010.
- [23] Katel. Introducing deep reinforcement learning. <https://www.katel.ro/en/semaforizare/14>, Oct 2018. Accessed on 2020-09-17.
- [24] Zawar H Khan and T Aaron Gulliver. A macroscopic traffic model for traffic flow harmonization. *European Transport Research Review*, 10(2):1–12, 2018.
- [25] Erling Kristiansen, Claude Loisy, et al. Road traffic monitoring by satellite. *ESABu*, 115:52–57, 2003.
- [26] Marcelo Lacortt, Moacir Kripka, and Rosana Maria Luvezute Kripka. Modelos matemáticos para otimização do tráfego urbano semaforizado. *TEMA (São Carlos)*, 14(3):359–372, 2013.
- [27] Yongfu Li and Dihua Sun. Microscopic car-following model for the traffic flow: the state of the art. *Journal of Control Theory and Applications*, 10(2):133–143, 2012.
- [28] Gil Lima Lial. *Controlo de tráfego*. PhD thesis, 2011.
- [29] C Manliguez. Generalized confusion matrix for multiple classes. 2016.
- [30] Maria Eugénia Graça Martins. Desvio padrão amostral. *Revista de ciência elementar*, 1(1), 2013.
- [31] MC.AI. Signalhead system - designing. <https://mc.ai/introducing-deep-reinforcement-learning/>. Accessed on 2020/10/06.
- [32] Sara Moridpour, Majid Sarvi, and Geoff Rose. Lane changing models: a critical review. *Transportation letters*, 2(3):157–173, 2010.
- [33] Seri Oh, Stephen G Ritchie, and Cheol Oh. Real-time traffic measurement from single loop inductive signatures. *Transportation Research Record*, 1804(1):98–106, 2002.
- [34] Mendes Oliveira. Sistema de monitorização para colmeias. 2015.

- [35] Duarte Paulo. <https://www.circulaseguro.pt/o-tempo-de-reacao-na-conducao/>, abr 2018. Accessed on 2020-09-03.
- [36] Rosana Cibely B Rego and Rodrigo S Semente. Sistema de controle de semáforo baseado na densidade de tráfego. *Anais do Encontro de Computação do Oeste Potiguar ECOP/UFERSA (ISSN 2526-7574)*, (1), 2017.
- [37] Daniel Shiffman. ml5: Friendly open source machine learning library for the web. <https://medium.com/ml5js/ml5-friendly-open-source-machine-learning-library-for-the-web-e802b5da3b2>, Jun 2018. Accessed on 2020-09-01.
- [38] Moira Turner and Lauren McCarthy. p5.js. <https://p5js.org/>. Accessed on 2020-09-03.
- [39] Femke van Wageningen-Kessels, Hans Van Lint, Kees Vuik, and Serge Hoogendoorn. Genealogy of traffic flow models. *EURO Journal on Transportation and Logistics*, 4(4):445–473, 2015.
- [40] SF Velosa and DD Pestana. Introdução à probabilidade e à estatística. *Lisboa: Fundação Calouste Gulbenkian*, 2010.