JOÃO GUILHERME MENDONÇA PIMENTA DE OLIVEIRA FERREIRA

PREVISÃO DE TRÁFEGO RODOVIÁRIO EM FUNÇÃO DAS CONDIÇÕES METEOROLÓGICAS

Universidade de Aveiro
2022

JOÃO GUILHERME MENDONÇA PIMENTA DE OLIVEIRA FERREIRA

PREVISÃO DE TRÁFEGO RODOVIÁRIO EM FUNÇÃO DAS CONDIÇÕES METEOROLÓGICAS

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor António Manuel Duarte Nogueira, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Pedro Alexandre de Sousa Gonçalves, Professor Adjunto da Escola Superior de Tecnologia e Gestão da Universidade de Aveiro.

o júri / the jury

presidente / president

Professor Doutor José Nuno Panelas Nunes Lau
professor associado da Universidade de Aveiro

vogais / examiners committee

Professor Doutor Joel Perdiz Arrais
professor auxiliar da Universidade de Coimbra

Professor Doutor António Manuel Duarte Nogueira
professor auxiliar da Universidade de Aveiro

agradecimentos / acknowledgements

Em primeiro lugar, gostaria de agradecer ao meu orientador, Professor Doutor António Nogueira, e ao meu coorientador, Professor Doutor Pedro Gonçalves, pela oportunidade de desenvolver a presente dissertação. Agradeço toda a dedicação, orientação e disponibilidade.

Ao Professor Doutor Fernando Braz e ao Professor Doutor Fabiano Baldo pela cooperação e ajuda prestada durante o desenvolvimento do presente trabalho.

À minha mãe, ao meu pai e ao meu irmão por me apoiarem durante todo este processo.

À Carolina por toda a ajuda, apoio incondicional, motivação e por nunca me ter deixado desistir, mesmo em tempos mais difíceis.

Aos meus amigos, Carina Neves, João Magalhães e José Moreira pela amizade, companheirismo e ajuda durante todo o meu percurso académico.

Palavras Chave Previsão, Tráfego rodoviário, Condições meteorológicas

Resumo

Observa-se que as condições meteorológicas condicionam a mobilidade, influenciando, por exemplo, os tempos de viagem e a segurança rodoviária. Nos últimos anos, o Instituto de Telecomunicações da Universidade de Aveiro tem vindo a desenvolver uma plataforma de comunicações veiculares. Foram instalados radares rodoviários que detetam a passagem de veículos para as praias da Barra e da Costa Nova, permitindo a contagem de veículos e análise de vários tipos de dados como, por exemplo, a velocidade e a hora de deteção. A conjugação destes dados com dados de outras fontes como, por exemplo, dados meteorológicos, permite efetuar previsões de tráfego como a quantidade de veículos ou o tempo de viagem. Contudo, estas previsões podem ser úteis noutras áreas como, por exemplo, avaliar a ocupação das redes de telecomunicações. Presentemente, a informação é serializada por um *broker* MQTT para uma base de dados relacional. Os objetivos desta dissertação compreendem o estudo das fontes de dados, técnicas de aprendizagem necessárias para a análise dos mesmos e desenvolvimento e teste de uma solução de previsão de dados, capaz de realizar a previsão do tráfego em tempo real de forma a disponibilizar aos automobilistas informação que lhes permita tomar decisões relacionadas com as viagens e assim evitar engarrafamentos e constrangimentos relacionados com deslocações em alturas de tráfego elevado.
Esta dissertação foi realizada com o apoio do Instituto de Telecomunicações.

Keywords Forecast, Road traffic, Weather conditions

Abstract

It is observed that weather conditions affect mobility, influencing, for example, travel times and road safety. In the last years, the Telecommunications Institute of the University of Aveiro has been developing a vehicular communications platform. Road radars were installed in order to detect the passage of vehicles towards the beaches of Barra and Costa Nova, allowing vehicles to be counted and analysis of various types of data, such as speed and detection time. The combination of this data with data from other sources, such as weather data, allows for traffic forecasts such as the number of vehicles or travel time. However, these forecasts can be useful in other areas, such as assessing the occupancy of telecommunications networks. Currently, the information is serialized by an MQTT broker to a relational database. The objectives of this dissertation comprise the study of data sources, learning techniques necessary for their analysis and development and testing of a data forecasting solution, capable of performing real-time traffic forecasting in order to provide motorists with information that allows them to make decisions related to travel and thus avoid traffic jams and constraints related to travel at high traffic times. This dissertation was carried out with the support of the Telecommunications Institute.

Conteúdo

Lista de Figuras

Lista de Tabelas

Glossário

ANN	Artificial Neural Network
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
ARLSTM	Autoregressive Long Short-Term Memory
BRANN	Bayesian Regularized Artificial Neural Networks
CNN	Convolutional Neural Network
CPNN	Counterpropagation Neural Network
CSV	Comma Separated Value
DFis	Departamento de Física da Universidade de Aveiro
ECMWF	European Centre for Medium-Range Weather Forecasts
ES	Exponential Smoothing
GB	Gradient Boosting
GPS	Global Positioning System
GRNN	General Regression Neural Network
IPMA	Instituto Português do Mar e da Atmosfera
IT	Instituto de Telecomunicações da Universidade de Aveiro
JSON	JavaScript Object Notation
LSTM	Long short-term Memory Recurrent Neural Network
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
ML	*Machine Learning*
MLP	Multi-Layer Perceptron
MLR	Multiple Linear Regression
NB	Naïve Bayes
RBFNN	Radial Basis Function Neural Network
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SMAPE	Symmetric Mean Absolute Percentage Error
SVM	Support Vector Machine
XGBoost	eXtreme Gradient Boosting

CAPÍTULO 1

Introdução

1.1 Contexto

O número de veículos em Portugal tem vindo a aumentar, destacando-se o número de veículos ligeiros [1]. O aumento do número de veículos conduziu a um aumento esperado do volume de tráfego. O volume de tráfego pode ser influenciado por diversos fatores como, por exemplo, condições meteorológicas ou trabalhos na via. As condições meteorológicas constituem um fator condicionante do quotidiano. A mobilidade é, consequentemente, um dos aspetos condicionados pelas condições meteorológicas, dado que, por exemplo, em condições meteorológicas adversas, a disposição para deslocações é reduzida. Esta diminuição do desejo de deslocação pode ser provocada pelas condições meteorológicas adversas e resultante da conjugação de inúmeros fatores como, por exemplo, o aumento do tempo de viagem ou questões de segurança.

Nos últimos anos, o Instituto de Telecomunicações desenvolveu uma plataforma de comunicações veiculares que recolhe, reencaminha e guarda dados das comunicações veiculares e dados rodoviários, instanciando um conjunto de radares rodoviários e sensores de estacionamento nas praias da Barra e Costa Nova [2]. Os radares rodoviários e sensores de estacionamento permitem a contagem dos carros que entraram/saíram destas praias, medição das velocidades, horas de passagem e estacionamento. Após a recolha destes dados e ligação com dados de outras fontes, como a informação meteorológica [3], é possível realizar previsões de tráfego, de ocupação de lugares de estacionamento, de tempo de viagem, entre outros, em função das condições meteorológicas, hora, dia da semana e período do ano. A previsão de tráfego envolve várias aplicações, das quais podemos destacar o tempo de viagem e a ocupação das vias de trânsito.

O exemplo mencionado no parágrafo anterior constitui uma abordagem simples e superficial. Contudo, as previsões de fluxos de tráfego, de tempos de viagem ou de ocupação de parques de estacionamento podem ser bastante frutíferas para a compreensão da ocupação das redes de comunicações e planeamento das atividades sociais.

A informação é serializada através de um Broker MQTT para um repositório implementado por uma base de dados relacional. Pretende-se que esta informação seja examinada, correlacionada com outras fontes de dados e utilizada como suporte de uma interface Web, facilitando o acesso por parte do utilizador.

1.2 Objetivos

O principal objetivo desta dissertação consiste na produção de uma solução capaz de prever os dados do tráfego em tempo real baseando-se nas condições meteorológicas recolhidas e nos próprios dados de tráfego, recorrendo a uma rede neuronal. Assim, esta dissertação compreendeu cinco fases:

- Estudo dos dados recolhidos provenientes dos radares e estações meteorológicas;
- Análise das técnicas de aprendizagem para a respetiva análise dos dados;
- Desenvolvimento e teste de uma solução para a previsão de tráfego em tempo real;
- Integração da solução com um motor de processamento de dados;
- Integração da solução com dashboard do grupo de comunicações veiculares.

Para além da introdução, esta dissertação seguirá a seguinte estrutura:

- Estado da arte - capítulo no qual é definido um enquadramento temático, enumerados estudos e projetos realizados, descritos problemas, analisados algoritmos entre outros;
- Trabalho desenvolvido - capítulo no qual é realizada a análise do *dataset* utilizado, do mecanismo de previsão e da interface de programação de aplicações;
- Testes e análise de resultados - capítulo que inclui a análise do impacto da COVID no fluxo das praias da Barra e da Costa Nova e comparação de vários algoritmos de previsão;
- Conclusões - capítulo no qual são demonstradas as conclusões desta dissertação e respetivas limitações.

1.3 Resultados

Após o desenvolvimento desta dissertação, concluiu-se que o algoritmo utilizado para a realização das previsões era a melhor opção dentro das hipóteses consideradas dado que era o algoritmo com menor Mean Absolute Error (MAE) e menor tempo de execução para a construção do modelo de dados.

Concluiu-se ainda que a utilização de dados meteorológicos conjugados com dados de tráfego para a realização de previsões de tráfego rodoviário era uma melhor opção em relação à utilização de dados exclusivamente de tráfego.

No decorrer desta dissertação foram escritos e publicados dois artigos científicos:

- J. Ferreira, F. Braz, F. Baldo e P. Gonçalves, «Analyzing the impact of the covid epidemic on beach access traffic,» em 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 2022, pp. 1–7. doi: 10.23919/CISTI54924.2022.9820252 [4];

- F. J. Braz, J. Ferreira, F. Gonçalves et al., «Road Traffic Forecast Based on Meteorological Information through Deep Learning Methods,» Sensors, vol. 22, n.o 12, 2022, issn: 1424-8220. doi: 10 . 3390 / s22124485. URL: https://www.mdpi.com/1424-8220/22/12/4485 [5];

CAPÍTULO 2

Estado da arte

2.1 Enquadramento

A previsão de fluxos de tráfego é uma técnica fundamental para a implementação de sistemas de transporte inteligentes – *Intelligent transportation systems*, sendo que estes sistemas têm vindo a ser estudados há mais de vinte anos [6]. O conceito de congestionamento no trânsito pode ser definido como uma condição nas vias de trânsito, podendo ser caracterizada por velocidades reduzidas e, consequentemente, maiores tempos de viagem [7]. Atualmente, nos grandes centros urbanos, a tentativa de resolução dos problemas provenientes do tráfego existente é constante. Inicialmente, para a previsão de tráfego, os algoritmos desenvolvidos baseavam-se em modelos autorregressivos e também noutros modelos de análise desenvolvidos para dados temporais [8]. Recentemente, surgiram métodos de *deep learning* que, dada a sua capacidade superior de previsão, têm sido considerados como sendo uma referência para a previsão de fluxos de tráfego, *natural language processing*, visão por computador e reconhecimento de voz [6]. Os algoritmos de *deep learning* desenvolvem as suas aptidões através de uma sequência de transformações não-lineares, criando relações entre as variáveis que, neste caso, seriam dados resultantes da análise de fluxos de tráfego. *Deep learning* torna-se, assim, um método eficaz para a previsão de fluxos de tráfego.

No presente, estabelecem-se três formas de atenuar os efeitos negativos resultantes do excesso de tráfego nos centros urbanos: aumentar as infraestruturas; promover alternativas de transporte como, por exemplo, o transporte público; e realizar uma gestão eficiente dos fluxos de tráfego. A gestão de fluxos de tráfego tem vindo a ser desenvolvida nas últimas décadas, resultante do aumento de dados recolhidos e também do avanço das tecnologias utilizadas para a recolha desses dados [8]. O desenvolvimento da gestão de fluxos de tráfego permite medir, modelar e interpretar as características do tráfego como, por exemplo, os tempos de viagem de cada veículo. Para além da gestão de fluxos de tráfego, os métodos de previsão também evoluíram a uma velocidade semelhante e, apesar de uma boa parte da pesquisa ser sustentada por análises de séries de dados temporais, também uma boa parte aborda métodos de *Machine Learning* (ML) [8]. Todavia, aspetos como o horizonte de previsão têm-se mantido

constantes por décadas. Quase toda a literatura é baseada em previsões com pouco alcance temporal [8]. Para além disso, a maioria dos estudos baseia-se em métodos de *deep learning* simples incapazes de reconhecer padrões de tráfego e correlacionar fatores externos (como as condições meteorológicas) com os fluxos analisados. Para além destes fatores, é possível definir outros que influenciam indiretamente o tráfego como, por exemplo, se o dia a ser analisado é um dia útil, a experiência de condução de cada condutor, se há eventos nas proximidades, entre outros [9].

Assim, o reconhecimento de congestionamento (ou falta dele) no trânsito é um elemento de decisão fundamental para vários momentos do quotidiano como, por exemplo, a realização de uma viagem [7]. Em [7], é afirmado que o congestionamento pode ser medido qualitativa ou quantitativamente e é também definida uma fórmula para o cálculo do valor de congestionamento – *Traffic Congestion Score*. Nesta fórmula os principais fatores são a velocidade, o tempo de viagem e o volume de tráfego.

Os dois principais métodos de previsão são os métodos estatísticos e os métodos de ML [6]. Foram definidas abordagens recorrendo a Support Vector Machine (SVM), Artificial Neural Network (ANN), métodos de *deep learning* e técnicas híbridas, combinando dois ou mais algoritmos [6]. Foram estabelecidos métodos que consistem em, inicialmente, escolher um conjunto de dados históricos de acordo com fatores de um determinado contexto como, por exemplo, o dia e as condições meteorológicas. Posteriormente, este conjunto de dados é utilizado no modelo de previsão, fornecendo uma estratégia eficaz para atingir a solução. De acordo com [6], estes métodos são denominados prediction-after-classification methods, sendo que, resumidamente, consistem em três etapas: classificação dos dados com base na similaridade dos dados, escolha de um grupo de dados adequado para o dia pretendido e, por fim, condução da previsão utilizando o grupo escolhido. Os autores começam por classificar os dados históricos, dividindo-os em dois grupos, de acordo com a sua similaridade. Após a classificação, separaram os fluxos de tráfego, resultando num grupo de fluxos de tráfego básicos e num grupo de fluxos de tráfego desviantes [6]. Nas previsões para apenas um dia, os dados históricos seriam escolhidos tendo em conta se o dia a prever é um dia útil ou não. A média dos valores dentro do grupo de dados históricos era considerada como sendo o resultado de previsão. Após este passo, o desvio do dia a ser previsto foi calculado através de métodos comuns para séries temporais, como Autoregressive Integrated Moving Average (ARIMA) e General Regression Neural Network (GRNN) algorithm. Para além do dia da semana, o fluxo de tráfego pode ser também afetado por outros fatores como as condições meteorológicas, como indicado pelo NCHRP Report 765 [6].

2.2 Estudos realizados

Ao longo dos tempos, foram desenvolvidas várias perspetivas para a previsão de tráfego. Assim, Van Arem *et al.* [8] pesquisou quais poderiam ser as aplicações de previsão de tráfego para a respetiva gestão, analisando o processo de um ponto de vista económico (procura e oferta). A procura representa os fluxos origem-destino e a oferta representa a capacidade da rede de estradas de satisfazer a procura. Em 2007, foi desenvolvido um estudo com

objetivo de perceber e examinar os modelos definidos anteriormente [8]. Os autores deste estudo consideraram que a previsão baseada em resultados anteriores ou na respetiva média, como é referido na secção 2.1 deste documento, como sendo pouco relevantes. Os critérios de comparação entre estes modelos foram o horizonte de previsão, âmbito da aplicação, velocidade computacional e precisão.

De acordo com o estudo realizado por [10], os métodos mais proeminentes são os não paramétricos, isto é, algoritmos que não fazem suposições fortes sobre a função de mapeamento, ou seja, a aprendizagem é livre a partir dos dados do conjunto de treino. Segundo os autores, alguns destes métodos podem incorporar dados ambientais como, por exemplo, as condições meteorológicas e, assim, aumentar a exatidão das previsões. De acordo com os autores, os métodos mais populares são CNN, LSTM e a combinação entre estes dois métodos.

2.3 Projetos previamente desenvolvidos

Em [7], é proposto um sistema de processamento de *Big Data* para criar um modelo de previsão de tráfego com base em condições meteorológicas. Este sistema é composto por duas partes, uma em que é utilizada a ferramenta Hadoop e outra em que é utilizada a linguagem de programação R. Na parte de Hadoop, os dados são processados.

Na parte de programação em R, é desenvolvido o modelo de previsão, recorrendo a uma Multiple Linear Regression (MLR). A análise recorrendo à MLR é composta por três fases.

A primeira fase consiste em criar um modelo de regressão completo. De seguida, é necessário remover as variáveis não influentes e as que têm elevada correlação com variáveis independentes. Após a criação do modelo, as variáveis que não são consideradas importantes são removidas à medida que o modelo é desenvolvido. Este método de remoção compreende três passos. O primeiro passo simplifica o modelo, removendo as variáveis independentes não necessárias. Com o modelo criado, o coeficiente de regressão das variáveis independentes é verificado, sendo que o valor extra representa o efeito de cada uma das variáveis independentes quando foram incluídas no modelo. Se esse valor não for significativo, a variável é removida do modelo e cria-se um novo modelo, até que o coeficiente de regressão fique estatisticamente significativo, isto é, só as variáveis independentes relevantes ficam no modelo. Dado que o número de variáveis continuava elevado (apenas dezoito foram eliminadas), foi necessário calcular a respetiva multicolinearidade entre as variáveis independentes. Um valor elevado para a multicolinearidade fez com que outras duas variáveis fossem removidas. Para terminar o processo de remoção, o terceiro passo consiste em calcular o *p-value*, isto é, a probabilidade significativa. Se o valor da probabilidade significativa for igual ou maior que 0.05, as condições meteorológicas não afetam o congestionamento do trânsito. Este valor foi utilizado de modo a que só fossem consideradas as variáveis relevantes.

Finalmente, deve ser realizada a análise residual, de modo a verificar a uniformidade e a normalidade da distribuição dos dados. O modelo MLR foi criado utilizando 48 variáveis de meteorologia e os dias da semana como sendo variáveis *dummy*. A precisão deste modelo é igual a 84.8%, confirmando a sua fiabilidade.

Em [6], é proposto um método inovador destinado à previsão do fluxo tráfego diário, recorrendo à mineração de padrões de fluxo de tráfegos de um ou vários dias, recorrendo a uma CNN. É extraída ainda informação útil, utilizando redes neurais profundas – *deep neural networks* [6]. Neste projeto os dados são inseridos num algoritmo de *deep learning* e incorporados com fatores externos, de modo a produzir a previsão pretendida. Dado que LSTM é um algoritmo amplamente recomendado por vários autores [6], foi escolhido como sendo o algoritmo para o modelo desenvolvido. Este método consiste em três etapas, começando por extrair os padrões dos dados inter e intradiários através da CNN. De seguida, é produzida a previsão de séries temporais utilizando LSTM e recorrendo aos padrões extraídos previamente. Por fim, o fluxo de tráfego é previsto.

A forma mais simples de prever o fluxo de tráfego diário consiste no cálculo da média dos valores recolhidos. Contudo, os dados recolhidos com uma duração considerável retratam padrões de fluxo que não podem ser reproduzidos e que variam de dia para dia. Em sentido contrário, as trajetórias em dias úteis apresentam similaridades [6].

Como mencionado anteriormente, *deep learning* é uma vertente de ML adequada para a previsão de fluxos de tráfego, o que conduziu os investigadores a adotarem esta técnica nos projetos desenvolvidos. Em [6], os autores referenciam um modelo de *deep learning* híbrido, utilizando uma CNN e Recurrent Neural Network (RNN), um algoritmo bastante poderoso para dados sequenciais, de acordo com [11]. O modelo proposto por [6] provou ser capaz de superar os habituais métodos estatísticos, como a média dos dados históricos. Contudo, os projetos baseados em *deep learning* foram desenvolvidos para intervalos de tempo a curto prazo, ou seja, prevêem os fluxos de tráfego para um futuro próximo [6].

Para além dos projetos já descritos, foi considerado também o modelo de previsão desenvolvido por [12]. Neste artigo científico, o condicionamento do tráfego resultante das condições meteorológicas é estudado, bem como a variância do tempo médio de viagem. De modo a analisar o impacto das condições meteorológicas no tráfego, os autores calculam o coeficiente de correlação, valor que varia entre -1 e 1. Por fim, os autores propõe um modelo de previsão de tráfego a longo-termo, baseado em algoritmos de aprendizagem.

Em [13] o tema é a previsão da utilização de estações de carregamento de veículos elétricos. Apesar de não estar diretamente relacionado com o tema da dissertação em questão, foi tido em consideração, dado que utiliza técnicas de ML conjugadas com fatores externos como, por exemplo, condições meteorológicas. Os autores deste artigo científico consideram que a limpeza e pré-processamento dos dados são duas etapas vitais para garantir a qualidade dos modelos de previsão, sendo que estas duas etapas incluem remover registos em falta e *outliers*. A presença de *outliers* pode provocar um impacto negativo no desempenho do modelo. De modo a combater os problemas originados pelos *outliers*, é comum utilizarem-se *boxplots* para os detetar.

O PASMO – Plataforma Aberta para o desenvolvimento e experimentação de Soluções para a Mobilidade é um projeto que visa desenhar e implementar uma plataforma de suporte à mobilidade inteligente. Este sistema inclui a recolha e transferência de dados e respetiva aplicação dos mesmos para estacionamento inteligente, análise das condições meteorológicas

que influenciam o volume de tráfego, contagem de veículos, entre outros [14].

No âmbito da mobilidade urbana, o PortoLivingLab é um sistema que recolhe dados de diversas fontes, recorrendo à *Internet of Things* de modo a analisar fenómenos meteorológicos, ambientais, de transportes públicos e fluxos humanos. Para a análise destes processos num contexto urbano, foi implementada uma rede veicular composta por mais de seiscentos veículos e dezanove sensores [15].

O PortoLivingLab define vários casos de uso como, por exemplo, a ligação WiFi nos autocarros públicos da cidade do Porto e os fluxos de passageiros, fornecendo novos indicadores da cidade [15].

2.4 Problemas e desafios

A previsão de tráfego é bastante desafiante, principalmente por ser influenciável por fatores complexos. Dado que os dados de tráfego são descritos ao longo do tempo e espaço, estão em constante mudança, tendo dependências espácio-temporais complexas e dinâmicas. Para além disso, estes dados são influenciados por fatores externos, como já foi anteriormente mencionado [9].

Embora a maioria da literatura esteja focada na previsão da acessibilidade dos recursos da estrada, o tempo de viagem é também uma variável a ser conjeturada, sendo que é mais percetível por humanos do que fluxos, velocidade e ocupação, características que podem ter diferentes interpretações de acordo com o tipo de estrada. Segundo o estudo descrito em [8], o principal defeito para a previsão do tempo de viagem é a falta de dados. De modo a superar este problema, os autores deste estudo propuseram a utilização de técnicas de simulação. Contudo, atualmente é possível prever tempos de viagem sem recorrer a métodos de simulação, dado que o número de dispositivos com Global Positioning System (GPS) tem proliferado, bem como o número de veículos conectados que podem fornecer dados sobre a viagem atual [8].

De acordo com [8], os autores salientaram a relevância das fontes de dados, dado que estudos com diferentes dados demonstram dificuldade em serem comparados. De acordo com este artigo científico, apenas alguns autores têm em consideração estes fatores. Em [8], é exibida uma tabela que prova que os critérios mais utilizados na literatura existente são aqueles que estão relacionados com os métodos de previsão, escala de previsão (localização específica, segmento da estrada ou a estrada completa) e variáveis de *output*. Lana et al. em [8] demonstram que a maioria dos modelos são construídos para contextos de autoestradas ou vias rápidas, sendo que o tráfego urbano é muito menos analisado e também demonstram que o horizonte de previsão é, na maioria do trabalho desenvolvido, inferior a uma hora.

Habitualmente, aumentar o horizonte de previsão não é visto como um desafio, dado que vários autores consideram que estender o horizonte de previsão conduz a uma deterioração das previsões. Contudo, a previsão a longo termo pode ser útil para sistemas avançados de gestão de tráfego, conhecidos como *advanced traffic management systems*. Até 2014, o horizonte de previsão dos projetos desenvolvidos é, maioritariamente, inferior a 60 minutos e é percetível que a maioria dos estudos mais recentes também definem horizontes de previsão

até 60 minutos. Contudo, é notável também um aumento do número de estudos com maiores horizontes de previsão, bem como o decréscimo dos modelos ARIMA e respetivas variantes [8]. Este decréscimo dos modelos ARIMA é resultante do facto destes modelos serem geralmente desenhados para *datasets* pequenos e não serem adequados para lidar com dados de séries temporais complexas e dinâmicas [9]. Para além disso, outra desvantagem dos modelos ARIMA é que estes apenas conseguem representar relações lineares entre as variáveis [16].

De acordo com [6], um dos maiores desafios da previsão de fluxos de tráfego está relacionado com a extração conjunta de dados intradiários e de dados interdiários, ou seja, a não extração exclusiva de dados interdiários.

Os modelos de previsão são frequentemente desenvolvidos com dados recolhidos apenas por uma fonte como, por exemplo, sensores, mas poderá ser difícil comparar dados recolhidos através de reconhecimento automático de alvos (*automatic target recognition*) com dados recolhidos por câmaras. Assim, técnicas de fusão de dados permitem combinar dados de diferentes fontes, sendo considerado por [8] como sendo um dos principais desafios na área da previsão de tráfego.

A combinação de técnicas de previsão é uma das predisposições existentes, e foi explorada inicialmente com a combinação de modelos ARIMA com outros métodos, de modo a aumentar a precisão e, mais recentemente, para otimizar o modelo [8]. Em [8], é demonstrado que a combinação de modelos com diferentes graus de complexidade espaço temporal com diferentes fatores exógenos é, provavelmente, a melhor escolha em termos de precisão.

Os fatores que afetam o tráfego e que não estão diretamente relacionados com o seu comportamento esperado são os principais obstáculos de uma previsão precisa de curto ou longo prazo. Utilizar estes fatores nos modelos de previsão pode melhorar o desempenho e é uma ação que deve ser considerada em trabalhos futuros [8]. Apesar de fatores externos como as condições meteorológicas afetarem o tráfego, segundo [17], a previsão do volume de tráfego com base nestes fatores constitui um desafio para os projetos a serem desenvolvidos. Para além disso, os autores consideram que os modelos existentes não toleram a ocorrência de alterações repentinas nas condições meteorológicas, o que resulta num desempenho fraco em condições extremas. Em [9] os autores consideram também que, durante o processo de recolha de dados de tráfego, devido a fatores como falhas do equipamento, os dados recolhidos são diferentes do seu valor real. Assim, os dados "contaminados", ao serem utilizados para a construção do modelo, irão afetar a respetiva exatidão.

Em [6], os autores afirmam que se nos debruçarmos sobre os fluxos de tráfego como uma sequência temporal convencional, é possível que alguns padrões interdiários possam ser perdidos. Para além disso, os autores afirmam também que alguns padrões intradiários podem não ser corretamente capturados.

De acordo com [18], as técnicas frequentemente utilizadas em ML são *clustering, feature extraction* e classificação. Segundo os autores, estamos perante uma lacuna na investigação existente, relacionada com a revisão sistemática das técnicas existentes, discussão de problemas associados a técnicas de ML, falta de sugestões para a resolução desses mesmos problemas e também na descrição de lacunas na investigação já realizada.

Um dos problemas detetados por [13] está relacionado com *feature engineering*, técnica que se refere à transformação de dados em representações com significado para humanos. Este processo é laborioso mas, segundo os autores, importante dado que é um das fraquezas dos algoritmos de aprendizagem. Um dos exemplos de *feature engineering* consiste na conversão do tempo em segundos para minutos e segundos, bastando para isso dividir os segundos por sessenta. O resto desta divisão corresponderia aos segundos.

Em [9] os autores consideram que, apesar da previsão de tráfego ter tido um grande progresso nos últimos anos, ainda existem vários desafios, para além dos anteriormente mencionados, que não foram completamente investigados. Nestes desafios, os autores incluem:

- a previsão em tempo real, afirmando que, devido ao elevado volume de dados, tamanho do modelo e número de parâmetros, o tempo de execução do algoritmo é bastante longo, impedindo as previsões em templo real;
- a interpretabilidade dos dados, considerando que, dada a complexidade dos dados, desenhar um modelo interpretável é uma tarefa mais desafiante do que para outros tipos de dados, como imagens e texto;
- a necessidade de estabelecer uma forma de comparação entre métodos, considerando ser difícil avaliar os modelos criados devido à falta de uma forma de os avaliar;
- escolha da melhor arquitetura de rede. Os autores consideram que, dada uma tarefa de previsão de tráfego, a forma como se escolhe uma arquitetura de rede adequada ainda não foi estudada de uma forma correta.

2.5 Algoritmos e métodos

Os algoritmos de ML são classificados em algoritmos de aprendizagem supervisionados ou não supervisionados. Nos algoritmos não supervisionados, o conjunto de dados de treino não é etiquetado e têm como objetivo agrupar *data points* similares. Em sentido contrário, nos algoritmos supervisionados, os modelos são treinados com um conjunto de dados etiquetados que contêm resultados específicos ou variáveis alvo, ou seja, a variável a ser prevista [13].

Dado que ML tem vindo a integrar soluções populares como, por exemplo, a previsão dos preços do petróleo, previsão do volume de vendas e previsão do comportamento de clientes, torna-se essencial definir modelos capazes de lidar com grandes volumes de dados complexos. Assim, vários métodos de agrupamento (*ensemble methods*) têm funcionado como uma ferramenta para a melhoria do desempenho dos modelos existentes [19]. Estes métodos baseiam-se, maioritariamente, em técnicas de randomização.

Recentemente, surgiu um novo método de agrupamento denominado eXtreme Gradient Boosting (XGBoost) [20], disponível em linguagens de programação populares como Python, R, Julia e integra pipelines de ciência de dados como, por exemplo, scikit-learn [19]. Este método é conhecido como um dos métodos mais precisos e com aprendizagem mais rápida, independentemente da natureza dos datasets que utiliza, como é evidenciado por [19]. Na referência [21] demonstram a eficácia deste método ao afirmarem que, na competição de ML do *website* Kaggle em 2015, em vinte e nove soluções vencedoras, dezassete utilizaram XGBoost.

Para além disso, demonstraram também que o sucesso deste método é comprovado na KDDCup 2015, competição na qual a totalidade das soluções vencedoras utilizaram XGBoost. Dado que a análise de dados desempenha um papel fundamental nas áreas de aplicação mencionadas no primeiro parágrafo desta secção, o facto de XGBoost ser uma escolha maioritária revela o seu impacto, importância e benefícios. Segundo [21], o fator mais importante por detrás do sucesso deste método é a sua escalabilidade em todos os cenários, dado que executa as suas operações dez vezes mais rápido do que outras soluções populares existentes. O XGBoost é um conjunto de árvores de decisão baseado em Gradient Boosting (GB). Um dos parâmetros a ser definidos no XGBoost está relacionado com a complexidade das árvores de decisão, que pode ser limitada utilizando estratégias como definir o tamanho máximo, entre outras. Definindo este parâmetro, os modelos são treinados mais rapidamente e necessitam de menos espaço computacional. O XGBoost foca-se em reduzir a complexidade computacional ao encontrar a melhor forma de separar os nós, o que é bastante vantajoso em relação a outros métodos, dado que a tarefa de separação é a tarefa mais morosa em algoritmos de construção de árvores de decisão. Para além disso, o XGBoost remove de forma eficaz os valores em falta [19]. Contudo, este método, baseado em GB, não é a única solução para obter resultados significativos. Assim, existem outras alternativas, como, por exemplo, Random Forest (RF).

Uma árvore de decisão pode ser utilizada para separar decisões complexas numa combinação de decisões mais simples utilizando pontos de separação dos atributos de *input*. Contudo, apesar de uma árvore de decisão ser simples de implementar, é propícia a *overfitting* (fenómeno no qual um modelo é bastante ajustado ao *dataset* utilizado, mas que se mostra ineficaz para prever novos dados). Para combater este problema, várias árvores de decisão podem ser agregadas, originando assim uma RF. RF consiste num conjunto de classificadores composto por árvores de decisão que são geradas utilizando duas fontes de randomização. É possível ajustar alguns dos parâmetros deste método, nomeadamente parâmetros relacionados com a profundidade das árvores de decisão. Habitualmente, as árvores de decisão crescem até que todos os nós folha sejam puros, i.e., até que todos os data points contenham a mesma etiqueta (*label*). Contudo, este crescimento pode produzir árvores com tamanhos consideráveis e, nestes casos, o tamanho de cada árvore pode ser reduzido, limitando a profundidade ou determinando um número mínimo de instâncias por nó antes ou depois da separação. Nos parâmetros que podem ser ajustados para uma RF, destacam-se o número máximo de atributos, o número mínimo de amostras necessárias para separar um nó, o número mínimo de amostras necessárias para criar um nó folha e a profundidade máxima da árvore [19].

Os algoritmos de boosting, como o GB e o XGBoost, combinam algoritmos de aprendizagem fracos (weak learners - algoritmos pouco melhores que algoritmos random) com algoritmos fortes, de uma forma iterativa. O GB é um algoritmo de boosting para o cálculo da regressão. Resumidamente, dado um dataset D = (x, y), o GB procura encontrar uma aproximação que mapeie os valores de x com os valores de y, minimizando o erro. Este método pode sofrer de *overfitting* se os seus processos não forem devidamente definidos. Contrariamente a RF, os valores pré-definidos para o GB são bastante limitados, nomeadamente no tamanho da árvore de decisão que, habitualmente, é limitado a três ou cinco níveis [19].

Em [19], é descrita uma comparação entre os métodos RF, GB e XGBoost. Para o teste destes métodos foram utilizados vinte e oito datasets do repositório UCI [22], provenientes de diferentes campos de aplicação, com diferentes números de atributos, classes e instâncias. Os métodos foram utilizados na sua forma pré-definida e também com os parâmetros ajustados. Os resultados deste estudo mostram que o método mais preciso foi o GB. Contudo, as diferenças para os valores do XGBoost e do RF nas suas versões predefinidas não são estatisticamente significantes, dado que são diferenças na ordem das décimas, por vezes, centésimas.

Osisanwo *et al.* [23] analisou sete algoritmos diferentes de ML: Decision Table, RF, Naïve Bayes (NB), SVM, Neural Networks (Perceptron), JRip e *Decision Tree*. Segundo os autores e tendo em consideração o dataset que foi utilizado (setecentas e oitenta e seis instâncias, oito variáveis independentes e uma variável dependente para análise), o SVM mostrou-se como sendo o algoritmo com maior precisão e exatidão. Esta afirmação é corroborada em [11], artigo no qual os autores afirmam que SVM obtém excelente desempenho em aplicações de previsão. Os algoritmos RF e NB foram considerados relevantes, dado que, em termos de precisão e exatidão, surgem imediatamente a seguir ao SVM. NB consiste em redes de *Bayes* bastante simples, compostas por grafos acíclicos apenas com um parente e vários filhos e baseiam-se em estimativas. SVM consiste nas mais recentes técnicas de algoritmos de aprendizagem supervisionados e baseiam-se num hiperplano que separa duas classes de dados. Este método, em conjunto com redes neuronais, tende a obter um desempenho aprazível quando necessita de lidar com atributos multi-dimensionais e contínuos. As principais qualidades deste método estão relacionadas com a exatidão, velocidade de classificação e tolerância a valores em falta. Comparando com RF, este método mostra-se mais rápido, com maior número de resultados corretos e com menor número de resultados incorretos. Estas propriedades verificam-se tanto para datasets considerados grandes (vários registos e múltiplos atributos) como para datasets considerados pequenos (pequenos registos e poucos atributos) [23].

No âmbito dos algoritmos utilizados para realizar previsões de valores, foi analisado um artigo científico cujo problema consiste na previsão da poluição do ar [24]. Neste artigo são utilizados dois algoritmos: RF e Multi-Layer Perceptron (MLP), que são, posteriormente, comparados. O MLP consiste numa rede neuronal sofisticada, composta por múltiplos perceptrons, sendo que possui três camadas: *input, output* e *hidden layers*. Os autores dividiram os dados em dois conjuntos (treino e teste), sendo que o conjunto de treino era composto por 60% do volume de dados e o de teste pelos restantes 40%. Neste artigo é também utilizada a matriz de confusão, conceito que será clarificado na secção 2.7 deste documento. De acordo com os autores, RF é uma melhor opção que MLP dado que obtém uma exatidão de 98% para o *dataset* utilizado e também um menor tempo de execução, tempo este que é resultante do facto de RF não provocar *overfit* dos dados. De modo a comprovar as afirmações descritas neste artigo, os autores optaram por utilizar outro dataset e voltaram a obter melhores resultados ao nível da exatidão para RF (88%), sendo que, no caso de MLP, a exatidão foi apenas 20%, um valor bastante baixo. Contudo, os autores também concluíram que, à medida que o número de neurons aumenta na camada *hidden* do MLP, este algoritmo aumenta a sua exatidão, enquanto que, à medida que o número de árvores de

decisão aumenta no RF, este diminui a sua exatidão. Assim, surge novamente a noção de que é estritamente necessário ajustar os parâmetros do algoritmo em utilização de modo a melhorar o seu desempenho, tal como foi referido anteriormente em relação ao XGBoost.

Em [11], os autores consideraram várias técnicas de previsão, em específico Bayesian Regularized Artificial Neural Networkss (BRANNs) e ANN. De acordo com os autores, as BRANNs evitam o fenómeno *overfitting* e são mais robustas, computacionalmente económicas e eficientes que as tradicionais ANN.

2.6 Redes neuronais

Dudek [16] considera diversas abordagens para realizar previsões baseadas em redes neuronais. Estas abordagens são posteriormente comparadas entre si. As técnicas consideradas são MLP, Radial Basis Function Neural Network (RBFNN), GRNN, Counterpropagation Neural Network (CPNN) e *self-organizing maps* (SOM). Uma das funcionalidades mais comuns destes métodos é a capacidade de aprendizagem a partir do reconhecimento de padrões, utilizando-os como *input* e, consequentemente, simplificando o problema em questão.

De acordo com [16], as redes neuronais são bastante utilizadas para problemas nos quais é necessária a previsão de valores. Esta forte utilização deve-se à flexibilidade das redes neuronais, que permite refletir a variabilidade de um determinado processo num ambiente dinâmico e também as relações, por vezes complexas, entre as variáveis em estudo.

Dada a importância dos algoritmos de previsão e respetiva complexidade, vários métodos foram desenvolvidos, podendo ser classificados em dois grupos: convencionais e não convencionais. Os métodos convencionais utilizam métodos de regressão, *smoothing techniques* e análise estatística, sendo os métodos mais comuns o método Holt-Winters Exponential Smoothing (ES) e os modelos ARIMA [16].

Por sua vez, os métodos não convencionais utilizam novos métodos computacionais como a inteligência artificial e ML, incluindo redes neuronais, *fuzzy inference systems, neuro-fuzzy systems*, SVMs, entre outros. Este métodos têm várias vantagens: adotam o teorema da aproximação universal (este teorema afirma que as redes neuronais são universais, isto é, existe sempre uma configuração de uma rede neuronal capaz de resolver o problema para que foi desenhada), capacidade de aprendizagem, paralelismo, robustez na presença de ruído nos dados e tolerância a falhas. Contudo, estes métodos também possuem algumas desvantagens: treino instável e disruptivo, apresentam dificuldades em coincidir a própria estrutura com a complexidade do problema, fraca capacidade de extrapolação e vários parâmetros para estimar (centenas de pesos) [16].

GRNN é um tipo de RBFNN adaptável a dados dispersos num espaço multidimensional. As vantagens deste método são a rápida aprendizagem e a fácil configuração de parâmetros [16].

Em suma, Dudek [16] concluiu que os modelos de redes neuronais que aprenderam recorrendo a padrões obtiveram um bom desempenho e que o modelo baseado em GRNN é o mais exato comparando com outros modelos de redes neuronais como MLP,RBFNN, SOM e CPNN e também com modelos estatísticos como ARIMA e ES. O GRNN é também o modelo

mais simples, tendo apenas um parâmetro para estimar. Este modelo é também o mais fácil de otimizar e bastante rápido.

2.7 Software desenvolvido

Para além dos estudos e projetos desenvolvidos mencionados anteriormente neste documento, atualmente existe *software* capaz de prever volumes de tráfego e simular viagens que os condutores pretendem realizar, independentemente do tipo de transporte que pretendem utilizar.

Um dos exemplos deste tipo de software é o Aimsun Next [25]. Este *software* simula a mobilidade em tempo real para antecipar o trânsito, combinado os dados históricos com dados em tempo real de autoestradas e cidades. Segundo os dados fornecidos pelos desenvolvedores deste produto, este sistema é utilizado, aproximadamente, por sete mil profissionais em noventa países, nomeadamente em cidades como Londres, Paris ou Barcelona.

Para além do Aimsun Next, foi desenvolvido também um tipo de software para a previsão de tráfego nos centros urbanos, apelidado de AnyLogic. Este *software* fornece a previsão de fluxos de tráfego, recorrendo a técnicas de simulação para atingir os resultados esperados. É utilizado para planeamento de tráfego, estudo das alterações nas vias de trânsito, sincronização de sinais luminosos, gerar estatísticas, entre outros [26].

2.8 Métricas de desempenho

De modo a testar o desempenho do modelo proposto, [17] recorreram a cinco métricas de desempenho: Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute Percentage Error (SMAPE), MAE, Mean Square Error e Root Mean Square Error (RMSE). Os autores consideraram que estas métricas refletem a diferença entre os valores reais e previstos, cumprindo o seu propósito. Em [12], é utilizado apenas RMSE. No artigo científico escrito por [9], os autores utilizaram MAE, RMSE e MAPE. Em [13], as métricas utilizadas para avaliar o desempenho dos modelos foram RMSE, MAE, SMAPE e também o coeficiente de determinação (R^2). Geralmente, valores pequenos de RMSE, MAE e SMAPE demonstram previsões exatas. O coeficiente de determinação é uma medida da qualidade do ajuste dos dados e encontra-se, habitualmente, entre zero e um. Se o valor do coeficiente de determinação for igual a um, estamos perante uma previsão perfeita. Geralmente, um maior valor do coeficiente de determinação demonstra um melhor desempenho [13]. Em [11], é utilizada apenas a métrica de desempenho Mean Absolute Scaled Error (MASE). Contudo, segundo estes autores, a falta de dados pode dificultar a produção de resultados, afetando o valor do MASE. De acordo com os autores, os dados recolhidos por sensores, como é o caso desta dissertação, são os mais sensíveis a falhas na recolha de dados, admitindo que esta falha pode ser resultante dos próprios sensores ou de problemas de conexão. Para combater este problema, foi utilizado um filtro de Kalman, um algoritmo que produz estimativas de dados desconhecidos, dadas as medidas observadas no espaço temporal.

Dada a abundância de métodos de previsão de tráfego, surgem também métricas de desempenho para cada método. Apesar de RMSE e MAPE serem métricas bastante utilizadas para a avaliação do desempenho de um modelo, não fornecem medidas comparáveis quando os modelos são divergentes como, por exemplo, uma rede neuronal e um ARIMA, ou quando os datasets de *input* dos modelos são completamente diferentes [8]. No caso de modelos *network-wide*, os erros podem propagar-se pelo tempo e espaço da rede e, portanto, uma correlação espaço-tempo entre previsões sucessivas pode ajudar a avaliar o desempenho. Apesar dos estudos contemporâneos utilizarem RMSE e MAPE para avaliar o desempenho, a necessidade de avaliar *datasets*, ambientes e métricas de desempenho continua presente, como identificado por [8].

Em [27], é estudado o problema de avaliação dos diferentes modelos de classificação que são utilizados em ML. É necessário avaliar estes modelos de modo a encontrar a solução ótima para os modelos de classificação gerados no processo de construção. Segundo os autores, existem diferentes medidas para avaliar o desempenho de um modelo, sendo que o critério mais utilizado consiste no cálculo da exatidão.

A classificação é uma das tarefas mais comuns em ML e baseia-se na procura de semelhanças entre objetos, determinadas pela análise das suas características. Num problema de classificação, o número de classes é previamente conhecido e limitado. Na avaliação dos métodos de classificação surge um conceito básico de falha. Se a classificação num caso específico prevê uma classe que é diferente da classe atual, então, estamos perante uma falha de classificação. Esta noção de falha resulta na fórmula da exatidão, que pode ser definida como o número de casos corretamente classificados sobre o número total de casos. Contudo, a exatidão não reflete as diferenças entre os tipos de erro, o que constitui uma desvantagem na utilização desta fórmula. Para além disso, a exatidão é dependente da distribuição da classe no *dataset* [27].

É necessário distinguir diferentes tipos de erro porque as consequências resultantes desse erro podem ser diferentes. Por exemplo, em medicina, se um sistema estiver encarregue de classificar um caso de cancro como positivo ou negativo, o resultado desta classificação leva a dois tipos de consequência. No caso do sistema identificar um doente positivo como negativo o erro é mais importante do que numa situação inversa, dado que os médicos não irão considerar que o paciente está doente e, consequentemente, não irão aplicar nenhum tratamento [27].

Esta distinção do tipo de erros introduz-nos ao conceito de matriz de confusão, uma matriz bidimensional, na qual cada linha corresponde a uma classe e números de registos dessa classe e cada coluna corresponde a resultados corretos ou incorretos. Na diagonal desta matriz encontram-se os exemplos corretamente classificados, enquanto que as restantes posições mostram os exemplos incorretamente classificados. Esta matriz permite uma melhor análise dos diferentes tipos de erros. Nesta matriz são possíveis quatro resultados diferentes: falsos negativos, falsos positivos, verdadeiros negativos e verdadeiros positivos. Falsos negativos e falsos positivos são classificações incorretas, enquanto que verdadeiros negativos e verdadeiros positivos são classificações corretas [27].

Com esta matriz surgem também os conceitos de exatidão (já definido anteriormente), taxa

de verdadeiros positivos, taxa de falsos positivos, taxa de verdadeiros negativos, taxa de falsos negativos e precisão. Consideremos um exemplo simples em que observamos apenas duas classes: positivos e negativos. A taxa de verdadeiros positivos é a proporção de casos positivos corretamente identificados sobre o total de casos positivos identificados correta e incorretamente. A taxa de falsos positivos é a proporção dos casos negativos incorretamente classificados como positivos sobre o total de casos negativos identificados correta e incorretamente. A taxa de verdadeiros negativos é a proporção de casos negativos identificados corretamente sobre o número de casos negativos identificados como negativos ou positivos. A taxa de falsos negativos é a proporção de casos positivos identificados como negativos sobre o número de casos positivos identificados como negativos ou positivos. Finalmente, a precisão é igual aos casos positivos identificados corretamente sobre o total de casos identificados como positivos (sendo negativos ou positivos) [27]. Considerando um exemplo simples de apenas com duas classes (Tabela 2.1), os conceitos de matriz de confusão, de exatidão, taxa de verdadeiros positivos, de falsos positivos, de verdadeiros negativos, de falsos negativos e precisão são definidos pelas seguintes expressões matemáticas:

Tabela 2.1: Matriz de confusão.

	Classe prevista	
	Positivos	Negativos
Positivos	X	Y
Negativos	Z	W

- Exatidão

$$\frac{\text{número de exemplos corretamente classificados}}{\text{número de exemplos classificados}} = \frac{X+W}{X+W+Z+Y}$$

- Taxa de verdadeiros positivos

$$\frac{X}{X+Y}$$

- Taxa de falsos positivos

$$\frac{Z}{Z+W}$$

- Taxa de verdadeiros negativos

$$\frac{W}{Z+W}$$

- Taxa de falsos negativos

$$\frac{Y}{X+Y}$$

- Precisão

$$\frac{X}{X+Z}$$

Em Novakovic *et al.* [27], existem casos em que a exatidão não é adequada como, por exemplo, no caso do número de casos negativos ser muito maior que o número de casos positivos. Se consideramos apenas duas classes e uma delas é significativamente mais pequena que a

outra, é possível obter valores elevados de exatidão se todas as instâncias forem classificadas como sendo da classe maior. Por exemplo, se tivermos uma amostra de mil casos, dos quais novecentos e noventa e cinco são negativos e apenas cinco são positivos. Se o sistema classificar todos como negativos, a exatidão irá ser igual a 99.5%, apesar do classificador ter errado a classificação de todos os casos positivos. Neste tipo de casos, a exatidão não é uma medida adequada. Em ML, a maioria dos classificadores assume igual importância para todas as classes. Em termos práticos, a atribuição de diferentes custos de classificações erradas é uma prática comum.

Outra das formas de testar o desempenho de uma classificação são os gráficos Receiver Operating Characteristic (ROC). Estes gráficos são compostos por representações bidimensionais nas quais o eixo das abcissas representa a taxa de falsos positivos e o eixo das ordenas representa a taxa de verdadeiros positivos. Assim, o ponto (0,1) corresponde ao classificador perfeito, dado que classifica todos os casos corretamente. Em sentido contrário, o ponto (1,0) indica um classificador que é incorreto para todas as classificações. Os gráficos ROC são gráficos nos quais a curva ou ponto é independente da distribuição da classe ou custo dos erros, contêm toda a informação da matriz de confusão e fornecem uma ferramenta visual para testar a capacidade do classificador em identificar corretamente casos positivos e negativos que foram incorretamente classificados. A área por baixo da curva de um gráfico ROC pode ser utilizada como medida da exatidão para várias aplicações.

Concluindo, o desenvolvimento mais relevante em relação à previsão de tráfego nos últimos anos está relacionado com o avanço das técnicas e tecnologias, aumento exponencial de Big Data e ML e também com o crescimento da disponibilidade de dados de tráfego de diversas fontes. Este desenvolvimento permitiu uma mudança de modelos de séries temporais para modelos orientados para dados.

Os problemas encontrados estão relacionados com o intervalo de previsão, o modelo mais adequado, as métricas de avaliação para cada modelo e os modelos híbridos para melhorar os modelos criados. Os desafios para o futuro relacionam-se com a capacidade de aumentar o intervalo de previsão, incorporar fatores exógenos e aumentar a capacidade do modelo a novos dados.

Os algoritmos de ML requerem um reajustamento dos respetivos parâmetros de modo a conseguirmos alcançar o seu melhor desempenho. Não deve ser considerado apenas o tempo de construção do modelo, mas também a precisão e a classificação correta. Assim, o melhor algoritmo de aprendizagem para um determinado *dataset* não garante a mesma precisão e exatidão para um *dataset* diferente. Contudo, a principal questão relacionada com a escolha de um algoritmo não está relacionada com a sua superioridade em relação aos demais, mas sim em que condições esse algoritmo supera o desempenho dos restantes. SVM, NB e RF são algoritmos que possuem elevada precisão e exatidão, independentemente do número de atributos e instâncias.

O facto da maioria dos veículos estarem equipados com dispositivos e tecnologia de alta qualidade e a possibilidade dos utilizadores enviarem dados através dos telemóveis (via GPS, por exemplo), são fatores que irão influenciar a previsão de tráfego. No âmbito destes

fatores, os protocolos de comunicação, que permitem aos veículos recolher e enviar dados, vão revolucionar esta área.

Após esta adaptação dos dados de cada ficheiro, os dados irão ser submetidos a um modelo recorrendo a uma rede neuronal e irão ser calculadas as previsões de tráfego com base nos dados de tráfego e meteorológicos recolhidos.

Após o término da testagem do algoritmo da rede neuronal, o sistema deverá ser integrado com a infraestrutura, de modo a fornecer os dados para a dashboard.

Para além dos fatores anteriormente mencionados, a recolha de dados a partir de redes sociais e o fornecimento voluntário de informação por parte de pessoas pode melhorar significativamente a adaptabilidade dos algoritmos desenvolvidos para a previsão de tráfego.

No âmbito desta dissertação, inicialmente, os dados são recolhidos por um conjunto de três radares, situados nas praias da Barra e Costa Nova, registando diversos valores como, por exemplo, a velocidade de cada veículo. Cada um destes registos tem associado um timestamp e foi adicionado a um ficheiro Comma Separated Value (CSV). São considerados ainda outros dois ficheiros CSV, que contêm dados relativos ao tráfego e à meteorologia. O ficheiro com os dados dos radares tem registos sempre que um veículo ativa o radar (independente do tempo de intervalo), o ficheiro com os dados meteorológicos tem dados de dez em dez minutos e o ficheiro com os dados relativos ao tráfego possui dados de hora a hora. Dado esta discrepância de intervalos de valores, foi necessário realizar uma fusão dos dados, de modo a que ficassem todos com intervalos de valores semelhantes, neste caso, de dez em dez minutos. De modo a fundir os dados disponíveis para cada intervalo de dez minutos foi calculada a respetiva média desse intervalo.

CAPÍTULO 3

Trabalho desenvolvido

Neste capítulo é descrito o trabalho desenvolvido no âmbito desta dissertação, expondo o processo de construção do mecanismo de previsão em tempo real e a respetiva arquitetura. Adicionalmente, é apresentado o diagrama de interações entre as diferentes entidades e o processo inerente ao processamento de dados meteorológicos e de tráfego rodoviário. Finalmente, é descrita a interface de programação de aplicações.

3.1 Análise e caracterização do *dataset*

De modo a tornar possível a construção de um modelo de dados eficaz, foi analisado um *dataset* cujo tamanho excede os 50 GB e contém mais de 170 milhões de registos. Após a respetiva análise, este *dataset* foi preparado. A preparação inclui a remoção de *outliers*, remoção de registos incompletos e separação dos dados de acordo com o tipo de registos. Nos registos analisados, mais de 1 milhão corresponde ao ano 2019, mais de 137 milhões correspondem ao ano 2020 e mais de 31 milhões correspondem ao ano 2021. Este *dataset* contém 3 colunas: *tenant_id*, *queue* e *data*. A coluna *tenant_id* corresponde ao tipo de registo e pode ter os seguintes valores: *pasmo parking* (registo de estacionamento), *pasmo radars* (registo de radares), *pasmo ria radar* (registo de radares), *pasmo sonars* (registo de sonares), *pasmo vehicles* (registo de deteção de veículos) e *pasmo weather* (registo das condições meteorológicas). Após a análise e preparação dos dados, o número de registos era superior a 155 milhões.

A análise destes dados permitiu concluir que existe uma diferença considerável de volume de tráfego entre os meses da Primavera e Verão (Maio a Agosto) e os meses de Inverno. Para além disso, estes dados mostram também uma diferença considerável entre dias úteis e fins de semana. No caso do mês de Agosto, observou-se que existe um aumento do volume de tráfego entre as 10 e as 15 horas. No caso dos meses de Inverno, este aumento é praticamente inexistente. Este comportamento pode ser justificado pelo facto das Praias da Barra e da Costa Nova serem zonas de habitação nos meses de inverno e zonas de férias nos meses de verão.

Além disso, a análise destes dados confirma a diferença entre os vários meses do ano como, por exemplo, os Sábados de Agosto que apresentaram mais de 3500 passagens e as Segundas-feiras de Janeiro que apresentaram menos de 250. Também é possível identificar grandes diferenças entre semanas do mesmo mês, especialmente em Agosto, mês no qual se nota uma diferença entre as primeiras 2 semanas e as 2 últimas (menos passagens nas 2 últimas semanas) [4].

3.2 Construção do mecanismo de previsão em tempo real

3.2.1 Arquitetura

Analisando [4], determina-se que o tráfego rodoviário é afetado por vários fatores como, por exemplo, as condições meteorológicas e epidemias/pandemias. Assim, e como referido anteriormente, o objetivo desta dissertação consiste na produção de uma solução capaz de prever os dados do tráfego na extensão da Praia da Barra e da Costa Nova, baseando-se nas condições meteorológicas e também em dados de tráfego rodoviário, recorrendo a uma rede neuronal. Estas zonas balneares apresentam um fluxo de tráfego bastante específico e, consequentemente, não representam a mobilidade humana na totalidade. Estas áreas apresentam variações aos fins-de-semana e também por estações do ano, como evidenciado por [4].

Para o desenvolvimento desta dissertação foram considerados diversos algoritmos de previsão recorrendo a redes neuronais (CNN, LSTM, ARLSTM). Após a realização de testes relacionados com a performance de cada um dos algoritmos considerados, determinou-se que o algoritmo com melhor performance para o tipo de dados utilizado e nas condições em que iria ser utilizado seria o algoritmo correspondente a uma CNN, como evidenciado por [5]. Concluiu-se que este seria o algoritmo mais benéfico dado que possui o menor MAE e menor tempo para a construção do modelo de dados pretendido [5].

Assim, o projeto desenvolvido compreende quatro elementos:

- Componente para recolha de dados meteorológicos;
- Componente para recolha de dados de tráfego rodoviário recorrendo a radares;
- Rede neuronal - CNN, utilizada para o cálculo das previsões de tráfego rodoviário;
- Application Programming Interface (API) - a API é utilizada para distribuir as previsões realizadas;

3.2.2 Diagrama de interações

O projeto desenvolvido segue o seguinte diagrama de interações, presente na Figura 3.1.

Analisando o diagrama, tornam-se implícitas as seguintes operações:

1. Os dados meteorológicos são recolhidos recorrendo a uma máquina presente no DFis e enviados para um *broker* MQTT cujo endereço é `ccam.av.it.pt`, em intervalos de 10 minutos;
2. Os dados de tráfego rodoviário são recolhidos, tratados e guardados numa máquina remota presente no IT recorrendo a três radares instalados;

3. Ambos os tipos de dados são fundidos num único ficheiro CSV (futuramente utilizado na CNN);
4. A CNN reconstroi o modelo de dados a cada sete dias. Posteriormente, guarda o modelo mais recente num ficheiro .sav;
5. A API desenvolvida exibe os resultados das previsões em formato JavaScript Object Notation (JSON) (ou seja, existe uma conexão entre a API e o ficheiro no qual está guardado o modelo de dados);

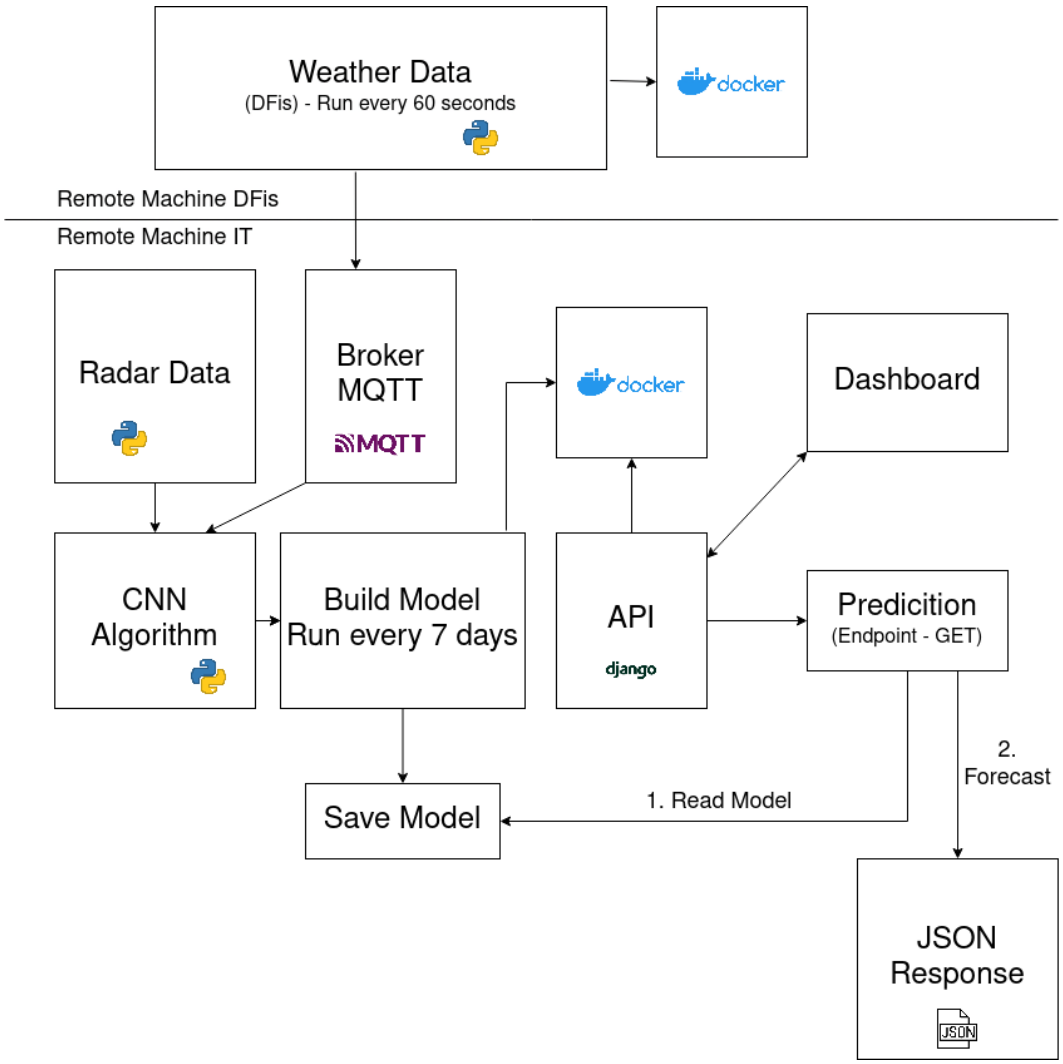


Figura 3.1: Diagrama de interações

3.2.3 Processamento de dados meteorológicos

De modo a recolher os dados meteorológicos necessários para sustentar o algoritmo de previsão utilizado, foram consideradas duas possíveis fontes de dados: Instituto Português do Mar e da Atmosfera (IPMA) e DFis. Os dados publicados pelo IPMA, fornecidos através da respetiva API, demonstraram uma capacidade de atualização baixa, não sendo apropriados para um modelo de previsão desenvolvido para calcular previsões em tempo real de forma

eficiente. Assim, esta fonte de dados foi descartada, considerando-a apenas para efeitos de teste. Em suma, a fonte definida para os dados meteorológicos foi o DFis.

O DFis recolhe os dados através de uma estação meteorológica, registando todos os valores num ficheiro XLSX, em intervalos de dez minutos, de uma hora e de um dia. O ficheiro XLSX é guardado numa máquina remota. Dado que os intervalos utilizados na CNN são equivalentes a blocos de dez minutos, apenas foram considerados intervalos deste tamanho, ignorando os restantes. Apesar da utilidade evidente destes dados, nem todos são relevantes para o cálculo da previsão, sendo necessário realizar um aperfeiçoamento dos mesmos, de modo a remover os dados em excesso e também de modo a formatá-los para o formato a ser utilizado na rede neuronal.

Cada registo dos dados meteorológicos, possui cinco colunas que o identificam: ano, mês, dia, hora e minuto. Os dados utilizados no modelo de previsão são:

- temperatura média, máxima e mínima em graus Celsius;
- rumo vento médio e máximo em graus;
- intensidade do vento média e máxima em m/s;
- precipitação em milímetros;
- radiação solar em KJ/m^2;

Assim, foram desenvolvidos dois *scripts* em Python de modo a ser possível produzir um conjunto de dados estruturados e devidamente formatados. O primeiro dos *scripts* é executado na máquina remota do DFis, sendo que o segundo é executado numa máquina remota presente no IT.

O primeiro script é responsável por ler o registo meteorológico mais recente (isto é, a última linha do ficheiro XLSX), recolher os valores necessários e, finalmente, enviar esse registo para o *broker* MQTT. Apesar de apenas existirem novos registos a cada dez minutos, este *script* é executado a cada sessenta segundos de modo a prevenir possíveis perdas de dados (por exemplo, o *script* consulta o ficheiro que contém os dados e este ficheiro ainda não foi atualizado). Se o *script* executasse a cada dez minutos e sucedesse a situação na qual o *script* consulta o ficheiro e este ainda não está atualizado, então estaríamos perante um problema no qual havia perda de registos, dado que só iria ser enviado o último registo. Assim, este *script* foi desenvolvido para consultar o ficheiro a cada sessenta segundos, minimizando a possível perda de dados.

Para cada registo enviado é guardada a data correspondente (ano, mês, dia, hora e minuto). De modo a evitar que sejam enviados registos repetidos, isto é, o último registo já foi enviado anteriormente e não ocorreu nenhuma atualização do ficheiro, a data do registo que está prestes a ser enviado é comparada com a data do último registo e, no caso de ser igual ou mais antiga, o registo é descartado. Caso contrário, é enviado.

Após obter o registo mais recente, é extraído o ano, mês, dia, hora, minuto, temperatura atual, rumo do vento, intensidade do vento, precipitação total e radiação solar. Finalmente, estes valores são enviados para o *broker*. O fluxo dos dados meteorológicos executado na máquina remota do DFis é descrito pela figura 3.2.

Posteriormente, os dados enviados a partir da máquina presente no DFis são recebidos numa máquina remota presente no IT. Inicialmente, o *script* guarda os valores recebidos

num ficheiro CSV auxiliar. Este ficheiro é utilizado de modo a ser possível calcular valores mínimos, máximos e médios.

Após registar os valores recebidos, os valores são agregados por ano, mês e dia, de modo a ser possível calcular o rumo médio do vento, a temperatura média, a temperatura máxima, a temperatura mínima, a intensidade média do vento, precipitação média e radiação solar média. Estes valores são, posteriormente, utilizados no algoritmo de previsão.

Após os cálculos dos valores mencionados no parágrafo anterior, o registo com os novos valores calculados é guardado num ficheiro CSV que posteriormente irá ser conjugado com os dados recolhidos através dos radares. O fluxo dos dados meteorológicos executado na máquina remota do IT é descrito pela figura 3.3. Ambos os *scripts* (DFis e IT) são executados em *Docker containers*, de modo a que executem de forma contínua.

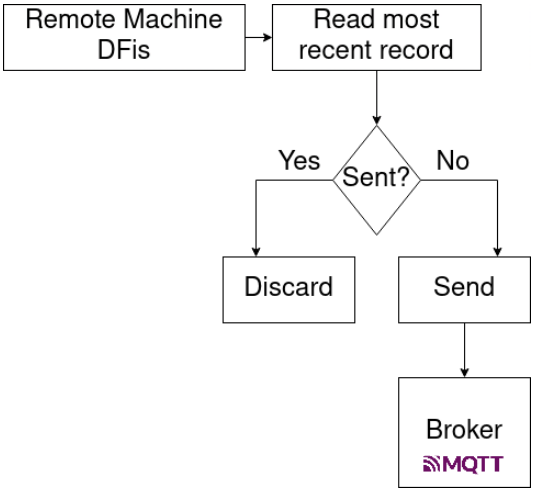


Figura 3.2: Fluxo de dados meteorológicos - Máquina DFis

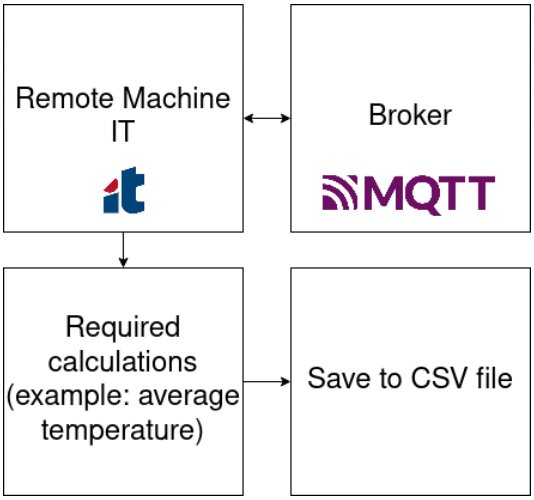


Figura 3.3: Fluxo de dados meteorológicos - Máquina IT

3.2.4 Processamento de dados de tráfego rodoviário recorrendo a radares

Desenvolveu-se um *script* de modo a recolher os dados de tráfego necessários para serem utilizados no algoritmo de previsão, provenientes dos radares presentes na auto-estrada A25 (ponte de acesso à Praia da Barra, Ílhavo) e também na Avenida José Estêvão (Costa Nova, Ílhavo). A localização destes radares é descrita pela figura 3.4. Este *script* é executado num *Docker container*, de modo a que execute de forma contínua.

Figura 3.4: Localização dos radares utilizados

Para recolher estes dados, este *script* começa por se conectar ao *broker* responsável pela receção dos registos provenientes dos dois radares disponibilizados. Cada mensagem recebida num determinado intervalo de 10 minutos é guardada num dicionário no formato { intervalo de 10 minutos : [registo1, registo2, ..., registoN] }. De forma contínua, quando ocorre uma mudança do intervalo de 10 minutos atual (por exemplo, o instante atual é 17:10:00, ou seja, início do segundo intervalo de 10 minutos das 17 horas), os registos do intervalo anterior são passados como argumento a uma função que irá extrair os valores necessários e, posteriormente, escrevê-los num ficheiro CSV. De seguida, os valores do intervalo de 10 minutos atual são continuamente agrupados e o processo repete-se.

Posteriormente, na função responsável pelos cálculos associados a este tipo de dados, são extraídos os valores da velocidade média, mínima e máxima para cada um dos radares em ambos os sentidos (entrada ou saída da zona balnear). É possível perceber qual o sentido de um determinado veículo através da respetiva velocidade vetorial (*xSpeed* e *ySpeed*). É também possível registar os veículos que foram detetados por cada radar, permitindo, assim, contar quantos veículos entraram e saíram destas zonas. A diferença entre os veículos que entraram e saíram de uma determinada zona é definida como sendo o fluxo. O fluxo das praias da Barra e da Costa Nova é também calculado. O valor do fluxo para cada uma destas praias é a variável a ser prevista pelo algoritmo de previsão. Finalmente, estes valores são guardados num ficheiro CSV para, posteriormente, serem conjugados com os dados recolhidos pela estação meteorológica e utilizados no algoritmo de previsão.

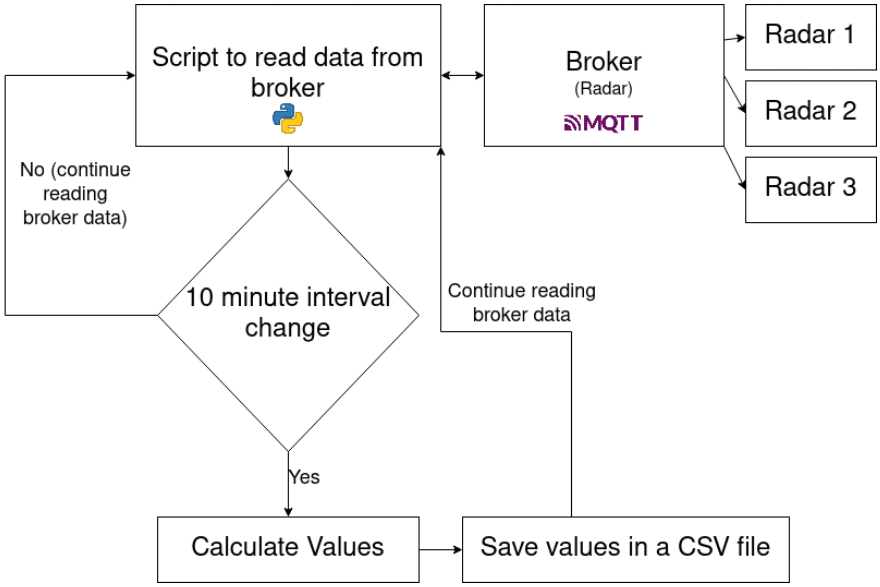


Figura 3.5: Fluxo de dados de tráfego rodoviário

3.3 Interface de Programação de Aplicações

A API desenvolvida permite que um utilizador tenha acesso às previsões realizadas pelo modelo de dados anteriormente desenvolvido via web. Permite também a respetiva integração noutras plataformas definidas previamente como, por exemplo, o *Traffic Control Center* [28]. A arquitetura da API é descrita pela figura 3.6.

O desenvolvimento da API incluiu três componentes: realização de previsões, integração em ambientes exteriores e respetiva *containerização*.

Assim, com o propósito de desenvolver a componente responsável pela realização das previsões (isto é, a componente lógica da API), utilizou-se a ferramenta Django [29]. Inicialmente, é estabelecido apenas um *endpoint* do tipo GET, que irá retornar as previsões pretendidas. Apesar de ser possível adicionar novos *endpoints* à API desenvolvida, não é necessário fazê-lo, dado que esta seria a única interação entre o utilizador e a mesma.

É possível integrar esta API em ambientes externos. De forma a concluir essa integração com sucesso, é suficiente realizar apenas chamadas ao *endpoint* fornecido, dado que este já devolve os resultados da previsão calculada.

Finalmente, para que a API esteja continuamente disponível via web foi necessário executá-la no interior de um *Docker container*. Assim, desenvolveu-se um *container* em Docker, perpetuando a disponibilidade da mesma.

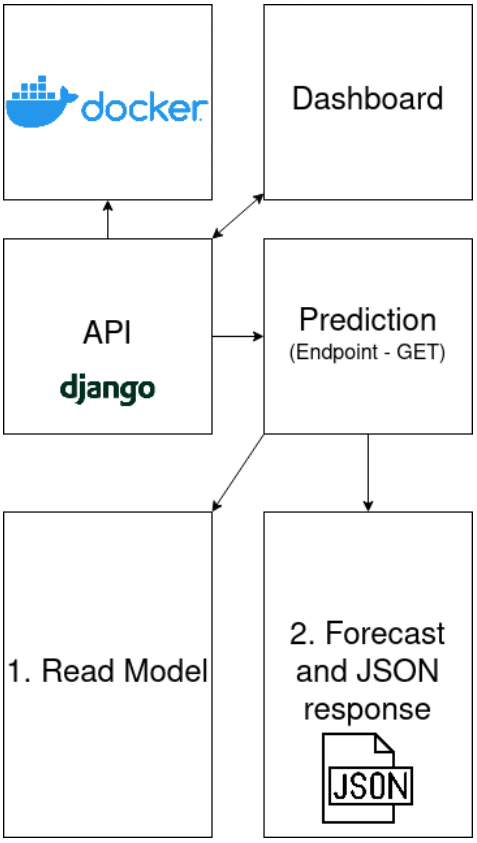


Figura 3.6: Interface de Programação de Aplicações

Por forma a desenvolver o mecanismo para o retorno do valor das previsões, foi criada a API descrita anteriormente. Após a criação desta API, foi acrescentado um *endpoint*, intitulado "prediction", cujo argumento imediatamente a seguir ao nome do *endpoint* corresponde ao tamanho do intervalo da previsão pretendido. Por exemplo, se o utilizador pretender obter uma previsão para os 30 minutos seguintes deverá efetuar uma chamada à API pelo seguinte endereço: `http://10.0.12.88:8080/predict/30`. A definição deste *endpoint* é imposta no ficheiro *urls.py*, presente no interior do directório resultante da criação da API.

De forma contínua, no momento da invocação deste *endpoint*, é, posteriormente, chamada a função *predict*, definida no ficheiro *views.py*, presente no interior do diretório resultante da criação da API. Inicialmente, esta função invoca uma das funções de uma classe desenvolvida, chamada CNN, na qual o algoritmo de previsão é construído e guardado. A função da classe CNN lê o modelo anteriormente criado e guardado e calcula a previsão para os 60 minutos seguintes (em intervalos de 10 minutos), retornando os valores relacionados com o tráfego existente para todos esses intervalos em formato de lista. De seguida, por uma questão de legibilidade, todos os valores retornados são arredondados às unidades.

Finalmente, é extraído o resultado pretendido dos resultados retornados pela classe responsável pelo algoritmo de previsão, isto é, se o utlizador pretender os 20 minutos seguintes

é extraído o segundo valor dos resultados retornados pelo algoritmo. Estes resultados são devolvidos ao utilizador em formato JSON, contendo:

- Velocidade média de entrada e saída de cada um dos três radares existentes (6 valores);
- Velocidade máxima de entrada e saída de cada um dos três radares existentes (6 valores);
- Velocidade mínima de entrada e saída de cada um dos três radares existentes (6 valores);
- Fluxo pretendido (diferença entre o número de carros que entraram e saíram numa determinada zona - Praia da Barra ou Costa Nova);

No total são retornados 19 valores, de acordo com a estrutura da figura 3.7.

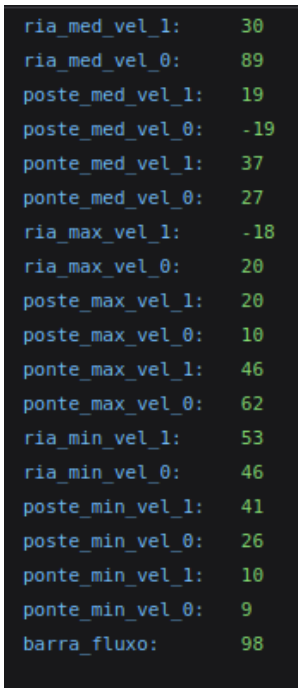


Figura 3.7: API - Resposta (JSON)

CAPÍTULO 4

Testes e análise de resultados

De forma a validar o sistema desenvolvido foram analisados e testados três algoritmos de previsão: CNN, LSTM e ARLSTM. Neste capítulo são descritos os testes efetuados aos algoritmos de previsão considerados, descrevendo o algoritmo utilizado. Adicionalmente, são apresentados e analisados os resultados obtidos, nomeadamente o MAE e o tempo de execução de cada algoritmo.

4.1 Análise da COVID no fluxo das praias da Barra e Costa Nova

Considerando que a pandemia da COVID-19 influenciou os fluxos de tráfego para as praias da Barra e da Costa Nova, identificou-se os eventos mais significativos desde Março de 2020 até ao fim do mesmo ano [4].

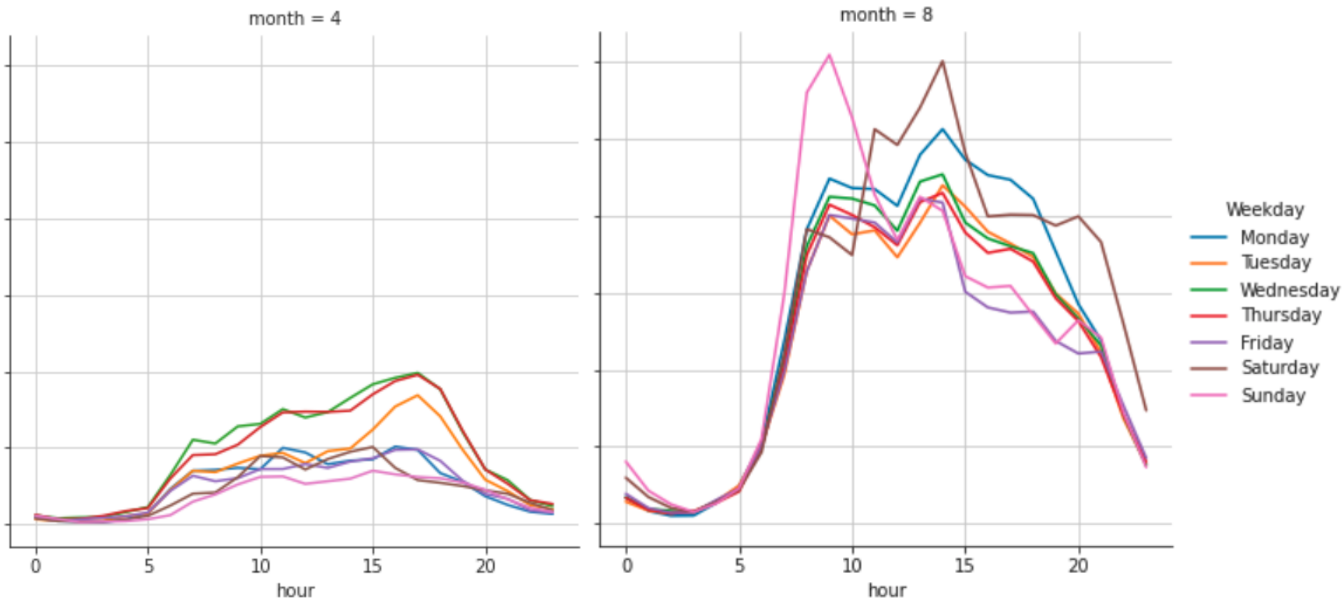
Os eventos registados estão descritos na tabela 4.1.

Data	Descrição
16/03/2020	Escolas e universidades encerram (transição para regime *online*).
19/03/2020	Confinamento de Março de 2020.
18/05/2020	Abertura de restaurantes e cafés; Início das aulas presenciais para o 11º e 12º ano.
30/07/2020	Abertura de bares e discotecas.
15/09/2020	Início das aulas e trabalho presencial.
29/10/2020	Proibição de transição entre concelhos.
31/10/2020	Início do recolher obrigatório entre as 23 e as 5h nos 121 municípios mais afetados. Aos fins de semana, o recolher obrigatório inicia-se à 1 hora.
12/11/2020	Fecho do comércio e restaurantes à 1h nos dois fins de semana seguintes.

Tabela 4.1: Restrições COVID-19 - Março a Dezembro 2020

A análise dos dados relativos ao tráfego permite perceber que em Abril de 2020 (isto é, durante o confinamento iniciado em Março do mesmo ano) o volume de tráfego foi reduzido (Figura 4.1a). Para além disso, é percetível a existência de uma diferença de volume de tráfego entre os meses da primavera e do verão (Maio até Agosto - Figura 4.1b) e os meses de inverno

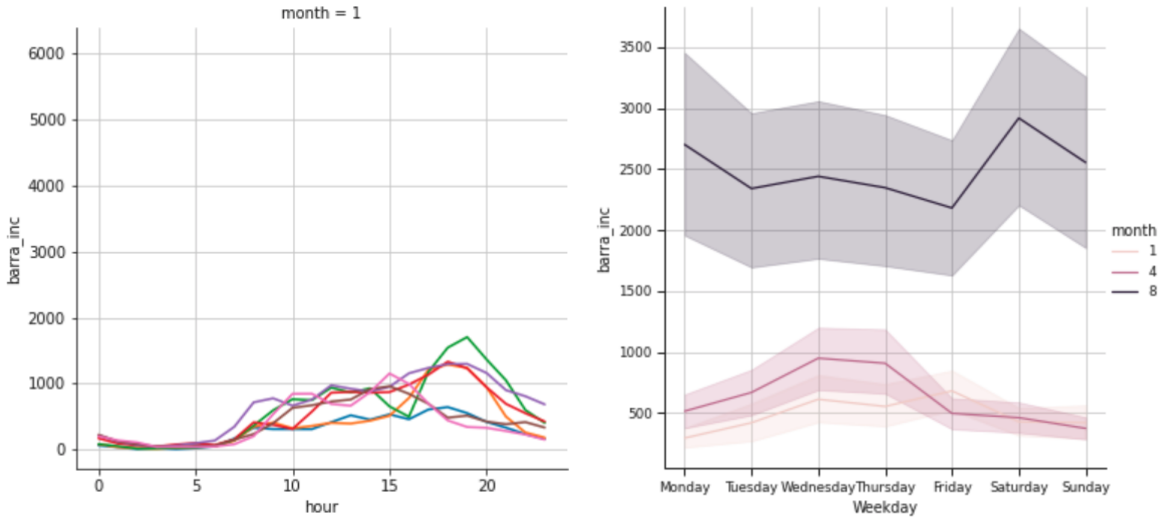
(Figura 4.2a) , bem como o aumento de tráfego aos fins-de-semana (Figura 4.2b). No fim deste período o número de veículos detectados aumentou, tornando impossível estabelecer uma relação entre este aumento e os eventos descritos anteriormente [4].



(a) Volume de tráfego em Abril de 2020

(b) Volume de tráfego em Agosto de 2020

Figura 4.1: Volume de tráfego em 2020



(a) Volume de tráfego em Janeiro de 2020

(b) Volume de tráfego por dia da semana

Figura 4.2: Volume de tráfego em 2020

4.2 Comparação de vários algoritmos de previsão

Durante o desenvolvimento desta dissertação, foram realizados testes com os dados recolhidos relativos às praias da Barra e da Costa Nova. Os testes foram executados numa máquina com processador Intel(R) Xeon(R) CPU E5-2620 v4 com 8 *cores*, com uma frequência de CPU igual a 2100 MHZ, memória RAM igual a 16GB e sistema operativo Ubuntu 20.04.4

(LTS). A máquina utilizada não possui interface gráfica ou outros serviços passíveis de perturbar o desempenho computacional. Os testes foram conduzidos remotamente via ssh. Foram utilizados intervalos de *input* e *output* equivalentes, desde 1 até 6, ou seja, para um intervalo de 10 minutos prever os 10 minutos seguintes, para um intervalo de 20 minutos prever os 20 minutos seguintes, entre outros.

Os algoritmos utilizados foram CNN, LSTM e LSTM. Os testes foram repetidos 10 vezes para cada intervalo, avaliando o MAE e o tempo de execução. Finalmente, a média e desvio padrão destas duas métricas eram calculados. Os resultados obtidos estão presentes nas tabelas 4.2, 4.3 e 4.4 .

Analisando a tabela 4.2, conclui-se que:

- para a praia da Barra, o teste com menor valor de MAE corresponde a intervalos de 10 minutos (*input* e *output* iguais a 1);
- para a praia da Barra, o teste com menor tempo de execução corresponde a intervalos de 10 minutos (*input* e *output* iguais a 1);
- para a praia da Costa Nova, o teste com menor valor de MAE corresponde a intervalos de 10 minutos (*input* e *output* iguais a 1);
- para a praia da Costa Nova, o teste com menor tempo de execução corresponde a intervalos de 60 minutos (*input* e *output* iguais a 6);

Prosseguindo a análise da 4.2, concluiu-se que o valor do MAE e o tempo de execução aumentam à media que o tamanho do *input* e *output* aumentam.

Praia	Input	Output	MAE (média)	MAE (desvio padrão)	Tempo de execução (média)	Tempo de execução (desvio padrão)
Barra	1	1	6,79	0,41	56,42	16,38
	2	2	6,81	0,41	63,42	14,86
	3	3	7,28	0,58	55,81	26,43
	4	4	7,31	0,55	60,14	17,59
	5	5	8,05	0,59	63,27	18,97
	6	6	8,47	0,53	66,22	29,52
Costa Nova	1	1	6,74	0,23	59,40	16,87
	2	2	6,88	0,47	66,62	22,93
	3	3	6,96	0,45	70,03	18,63
	4	4	7,53	0,53	55,54	19,20
	5	5	7,73	0,32	62,59	18,89
	6	6	8,79	0,36	54,19	12,88

Tabela 4.2: Resultados - CNN

A análise da tabela 4.3 permite verificar que:

- para a praia da Barra, o teste com menor valor de MAE corresponde a intervalos de 30 minutos (*input* e *output* iguais a 3);
- para a praia da Barra, o teste com menor tempo de execução corresponde a intervalos de 10 minutos (*input* e *output* iguais a 1);
- para a praia da Costa Nova, o teste com menor valor de MAE corresponde a intervalos de 10 minutos (*input* e *output* iguais a 1);

- para a praia da Costa Nova, o teste com menor tempo de execução corresponde a intervalos de 60 minutos (*input* e *output* iguais a 1);

Dado que os testes para intervalos de 10 minutos apresentam os menores valores de MAE e tempo de execução no caso da Praia da Costa Nova e menor tempo de execução no caso da Praia da Barra, é possível afirmar que esta seria o melhor intervalo. Poderiam ser consideradas outras opções como, por exemplo, intervalos de 30 minutos, dado que esta opção apresenta um valor de MAE inferior.

Praia	Input	Output	MAE (média)	MAE (desvio padrão)	Tempo de execução (média)	Tempo de execução (desvio padrão)
Barra	1	1	14,91	1,41	920,8	108,14
	2	2	13,84	2,39	1038,09	125,18
	3	3	13,71	1,94	1190,37	142,13
	4	4	13,83	1,61	1370,07	180,46
	5	5	13,94	3,35	1661,48	267,81
	6	6	14,73	1,17	1872,14	264,06
Costa Nova	1	1	13,49	1,88	991,62	101,98
	2	2	14,29	1,73	1080,23	132,31
	3	3	14,29	1,92	1188,94	203,84
	4	4	14,47	2,07	1399,33	143,40
	5	5	14,92	1,16	1744,69	168,46
	6	6	14,97	1,23	1641,91	268,67

Tabela 4.3: Resultados - LSTM

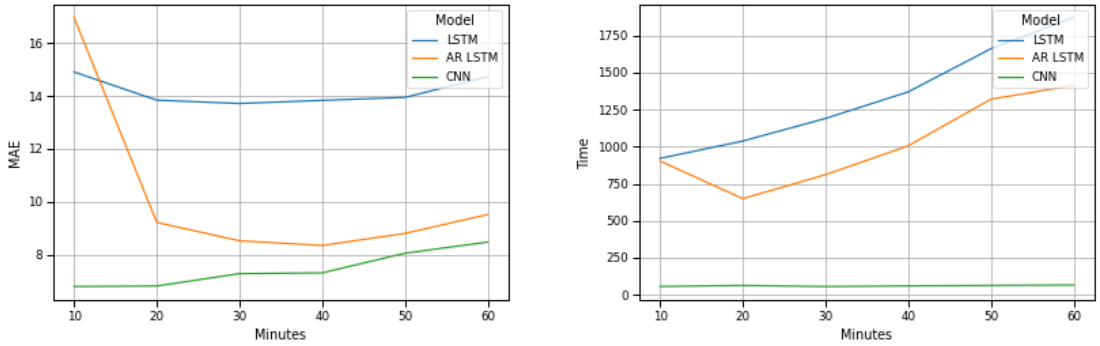
A tabela 4.4 permite concluir que:

- para a praia da Barra, o teste com menor valor de MAE corresponde a intervalos de 40 minutos (*input* e *output* iguais a 4);
- para a praia da Barra, o teste com menor tempo de execução corresponde a intervalos de 20 minutos (*input* e *output* iguais a 2);
- para a praia da Costa Nova, o teste com menor valor de MAE corresponde a intervalos de 30 minutos (*input* e *output* iguais a 3);
- para a praia da Costa Nova, o teste com menor tempo de execução corresponde a intervalos de 20 minutos (*input* e *output* iguais a 2);

Praia	Input	Output	MAE (média)	MAE (desvio padrão)	Tempo de execução (média)	Tempo de execução (desvio padrão)
Barra	1	1	16,97	4,67	902,99	48,63
	2	2	9,21	1,42	649,23	146,02
	3	3	8,52	0,57	811,02	121,65
	4	4	8,34	0,35	1006,01	96,47
	5	5	8,8	0,9	1320,48	218,51
	6	6	9,51	1,6	1413,75	370,97
Costa Nova	1	1	17,89	6,14	898,62	65,68
	2	2	8,76	0,63	730,98	154,28
	3	3	8,74	1,2	891,32	129,97
	4	4	8,66	0,95	1014,53	105,89
	5	5	9,29	1,02	1254,1	265,74
	6	6	11,92	2,67	1386,19	298,42

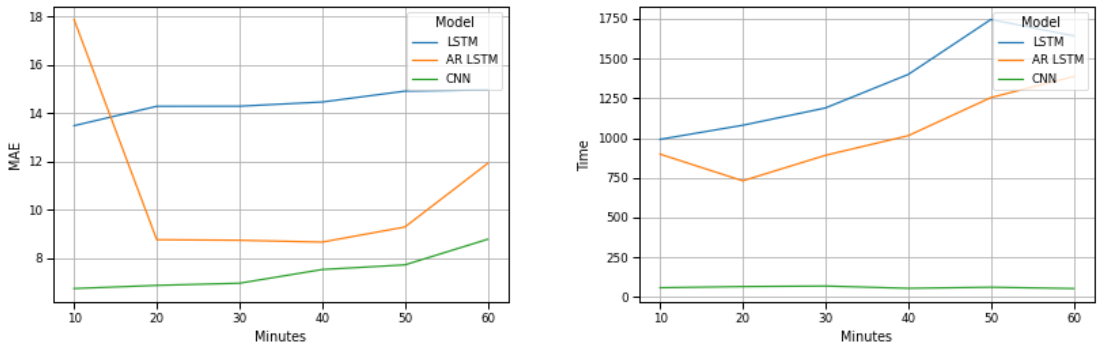
Tabela 4.4: Resultados - ARLSTM

Observando os resultados presentes nas tabelas 4.2, 4.3 e 4.4 e descritos nos gráficos 4.3a, 4.3b, 4.4a e 4.4b as vantagens de utilizar o algoritmo CNN para desenvolver um mecanismo de previsão de tráfego em tempo real são percetíveis, dado que este método é o que possui melhor desempenho em comparação com os restantes (menor MAE e menor tempo de execução) [5]. Assim, este foi o algoritmo escolhido para ser utilizado nesta dissertação.



(a) MAE **(b)** Tempo de execução

Figura 4.3: Resultados - Praia da Barra



(a) MAE **(b)** Tempo de execução

Figura 4.4: Resultados - Costa Nova

4.3 Algoritmo de previsão

Em função da comparação dos algoritmos descrita nas tabelas 4.2, 4.3 e 4.4 foi escolhido o algoritmo CNN por motivos de precisão e tempo de execução. Para a definição deste algoritmo e consequente modelo de dados é estabelecido um conjunto de fases que irão ser descritas nos parágrafos seguintes deste capítulo. De forma sucinta, é possível resumir este processo em quatro fases: preparação dos dados recolhidos, definição dos hiper-parâmetros, construção do modelo de dados utilizando os respetivos hiper-parâmetros e armazenamento local do modelo construído [30].

Inicialmente, desenvolveu-se uma função responsável pela preparação dos dados recolhidos. Esta função começa por extrair todos os atributos numéricos do conjunto de dados inicial, registando-os num novo conjunto de dados e preenchendo com o valor zero todas as colunas cujo valor não está definido. É necessário gerir os valores em falta dado que a performance de vários algoritmos de ML, como as CNN, é afetada. Para além disso, a gestão destes valores permite a criação de um modelo de dados mais robusto e facilita a utilização do respetivo algoritmo. Posteriormente, os atributos categóricos são também extraídos e registados num novo conjunto de dados. Finalmente, é extraída a coluna referente ao fluxo de tráfego existente (Praia da Barra ou Costa Nova) e registada numa nova variável. Após estas alterações, os

três novos sub-conjuntos de dados são concatenados, formando um novo conjunto de dados já formatados e adaptados para serem utilizados no algoritmo de previsão.

Subsequentemente, são estabelecidos os hiper-parâmetros a serem utilizados para a construção do modelo de dados. O conjunto de treino corresponde a 70% dos dados iniciais, o conjunto de validação corresponde a 20% dos dados iniciais e, finalmente, o conjunto de teste corresponde a 10% dos dados iniciais.

Posteriormente, é iniciada a construção do modelo de dados. Na função responsável pela construção do modelo de dados, são definidas duas variáveis essenciais para a construção do mesmo:

- *single step window* - consiste na definição do tamanho do intervalo de cada porção do resultado da previsão (neste caso, o tamanho seria equivalente a 10 minutos, ou seja, se o tamanho do resultado previsto fosse uma hora obtínhamos 6 *single step windows*);
- *baseline* - consiste na definição de um objeto simples que irá ser utilizado para efetuar comparações com o modelo construído.

De seguida, são definidos os tamanhos do intervalo de *input* e *output*. Os valores definidos foram 6 (60 minutos), para ambas as variáveis, com base nos resultados do trabalho desenvolvido no âmbito dos artigos científicos elaborados no decorrer desta dissertação [4] [5]. Os valores definidos são equivalentes a 6 dado que, para intervalos maiores, a qualidade da previsão é deteriorada (isto é, o valor do MAE aumenta), tornando a previsão pouco precisa [5].

Finalmente, o modelo de dados é construído, recorrendo à ferramenta TensorFlow. Após a construção do modelo, este é compilado e treinado (função *compile and fit*). O modelo construído é guardado localmente. Para além disso, é registada também a atual versão do modelo (valor inteiro que é incrementado a cada iteração). De modo a que todas as funções associadas à construção do modelo de dados proposto executem em intervalos de 7 dias, como referido anteriormente, o *script* associado é executado num *Docker container*.

4.3.1 Funções e classes auxiliares

Para além da metodologia mencionada na secção 4.3, foram ainda definidas funções e classes que auxiliam o processo de construção do modelo e fornecem suporte às interações existentes, nomeadamente com a API desenvolvida (descrita no capítulo 3). As funções definidas foram:

- Função *Predict* - Função utilizada para efetuar a previsão. Esta função é invocada pela API sempre que a última recebe um pedido GET. Inicialmente, carrega o modelo guardado em memória e, posteriormente, calcula a previsão, cujo tamanho corresponde aos 60 minutos seguintes (ou seja, os 6 intervalos de 10 minutos seguintes);
- Função *Adjust Prediction* - Função utilizada na avaliação do melhor modelo da CNN;
- Função *Compile and Fit* - Função utilizada para compilar o modelo (*compile*) para treino e, após compilado, treiná-lo (*fit*) seguindo um número de epochs previamente definido. Foram utilizadas 150 epochs e avaliado o MAE;
- Classe *Multi Column Label Encoder* - Transforma as colunas de um determinado *dataframe*, utilizando a função *fit_transform* do módulo *LabelEncoder()*. Esta transformação

resulta na conversão de colunas com valores categóricos em colunas com valores numéricos;

- Classe *Window Generator* - O modelo de dados utilizado produz um conjunto de previsões baseado num intervalo de tempo presente nos dados. Esta classe gere os intervalos de entrada e saída, define os intervalos de tempo resultantes, permite a criação de um gráfico ilustrativo com os resultados e gera porções de dados (*datasets*) extraídas dos conjuntos de dados de treino, avalição e teste.
- Classe *Baseline* - Função utilizada para construir um modelo simples (previsão para os 10 minutos seguintes, tendo em conta apenas um intervalo anterior equivalente a 10 minutos) que é comparado com o modelo futuramente criado.

4.3.2 Resultados - Recolha de dados meteorológicos

Como mencionado anteriormente, o processamento dos dados meteorológicos é composto por cinco etapas: ler o registo mais recente, enviar para o *broker*, ler os registos recebidos no *broker*, efetuar os cálculos necessários e guardar os valores obtidos num ficheiro CSV. A máquina responsável pela receção das mensagens utiliza Ubuntu 20.04.4 LTS como sistema operativo e possui 8 gigabytes de memória RAM, um disco duro (HDD) cujo tamanho é igual a 279 gigabytes e processador Intel(R) Xeon(R) com frequência equivalente a 2.10 gigahertz, com 4 núcleos. De modo a caraterizar o desempenho da solução desenvolvida, foram recolhidos os tempos de execução para a obtenção do registo mais recente, a duração do envio das mensagens e também o tempo necessário para efetuar os cálculos necessários. Foram realizadas mil experiências (aproximadamente uma semana de novos registos, isto é, um intervalo equivalente ao período entre as atualizações do modelo de dados construído), utilizando um ficheiro XLSX exemplo com registos com início no dia 1 de janeiro de 2021 e término no dia 14 de janeiro de 2022. Os resultados obtidos são descritos pela tabela 4.5.

Assim, para a obtenção do registo meteorológico mais recente os valores para os tempos de execução obtidos resultaram nas seguintes métricas de desempenho:

- Média equivalente a 6,52 segundos;
- Mediana equivalente 6,39 segundos;
- Desvio padrão equivalente a 0,24 segundos;

Estes valores seriam inferiores para um número de registos inferior, isto é, em janeiro o tempo necessário para obter o registo mais recente seria inferior em relação ao tempo necessário para a mesma tarefa realizada em dezembro do mesmo ano, dada a cardinalidade dos dados.

Analisou-se também a diferença entre o tempo de receção e de envio das mensagens. As métricas de desempenho recolhidas foram:

- Média equivalente a 67 milissegundos;
- Mediana equivalente a 50 milissegundos;
- Desvio padrão equivalente 103 milissegundos;

Em relação ao tempo de execução necessário para efetuar os cálculos resultantes dos valores recolhidos, os valores para os tempos de execução obtidos resultaram nas seguintes métricas de desempenho:

- Média equivalente a 33 milissegundos;
- Mediana equivalente a 32 milissegundos;
- Desvio padrão equivalente a 6 milissegundos;

Operação	Média	Mediana	Desvio Padrão
Obtenção do registo mais recente	6,52 s	6,39 s	0,24 s
Envio das mensagens	67 ms	50 ms	103 ms
Cálculos necessários	33 ms	32 ms	6 ms

Tabela 4.5: Resultados - Recolha de dados meteorológicos

4.3.3 Avaliação do mecanismo de previsão em tempo real - Dados meteorológicos e de tráfego rodoviário

Nesta secção são analisados os resultados provenientes do modelo de dados criado, nomeadamente os valores do MAE e tempo de execução durante a construção do mesmo. Os resultados obtidos são descritos pela tabela 4.6. De modo a testar a capacidade de previsão do modelo construído, o algoritmo desenvolvido foi executado 52 vezes (dado que o modelo seria reconstruído a cada 7 dias, 52 tentativas equivalem a um ano civil, aproximadamente). Para a construção do modelo pretendido, o algoritmo desenvolvido registou os seguintes valores para o tempo de execução:

- Média equivalente a 55,82 segundos;
- Mediana equivalente a 49,43 segundos;
- Desvio padrão equivalente a 19,43 segundos.

De modo a testar a capacidade de previsão do modelo construído, reuniram-se os seguintes valores de erro médio absoluto (MAE):

- Média equivalente a 8,52;
- Mediana equivalente a 8,51;
- Desvio padrão equivalente a 0,48;

CNN	Média	Mediana	Desvio Padrão
Construção do modelo	55,82 (s)	49,43 (s)	19,43 (s)
Erro médio absoluto	8,52	8,51	0,48

Tabela 4.6: Resultados - Algoritmo de previsão (CNN)

4.3.4 Avaliação do mecanismo de previsão em tempo real - Dados exclusivamente de tráfego rodoviário

Anteriormente, os dados utilizados para a construção do modelo de dados e realização das previsões pretendidas eram compostos por dados meteorológicos e dados de tráfego rodoviário. Contudo, de modo a testar a necessidade de utilizar estes dois tipos de dados de forma concatenada, foi desenvolvido um modelo de dados recorrendo apenas a dados de tráfego rodoviário. Assim, nesta secção são analisados os resultados provenientes do modelo de dados criado, nomeadamente os valores do MAE e tempo de execução durante a construção

do mesmo. De modo a testar a capacidade de previsão do modelo construído, o algoritmo desenvolvido foi executado 52 vezes, de forma semelhante ao exemplo anterior.

Para a construção do modelo pretendido, o algoritmo desenvolvido registou os seguintes valores para o tempo de execução:

- Média equivalente a 67,83 segundos;
- Mediana equivalente a 66,91 segundos;
- Desvio padrão equivalente a 12,13 segundos.

De modo a testar a capacidade de previsão do modelo construído, reuniram-se os seguintes valores de erro médio absoluto (MAE):

- Média equivalente a 10,89;
- Mediana equivalente a 10,82;
- Desvio padrão equivalente a 0,76;

Em suma, os resultados obtidos são descritos pela tabela 4.7.

	Dados de tráfego e dados meteorológicos		Dados de tráfego	
	MAE	Tempo de execução	MAE	Tempo de Execução
Média	8,52	55,82 (s)	10,89	67,83 (s)
Mediana	8,51	49,43 (s)	10,82	66,91 (s)
Desvio Padrão	0,48	19,43 (s)	0,76	12,13 (s)

Tabela 4.7: Resultados obtidos para ambos os testes efetuados

Analisando os dados presentes na tabela anterior, as vantagens da utilização dos dois tipos de dados (meteorológicos e de tráfego) em relação à utilização exclusiva de dados de tráfego são percetíveis, podendo afirmar-se que:

- o modelo de dados construído recorrendo a dados meteorológicos e de tráfego é mais preciso que o modelo de dados construído recorrendo apenas a dados de tráfego, dado que o primeiro obtem menor MAE (menor média e menor mediana). Também é possível afirmar que o primeiro agrupa valores mais aproximados entre si (menor desvio padrão);
- o tempo de construção do modelo de dados construído recorrendo a dados meteorológicos e de tráfego é menor, portanto, mais rápido em relação ao tempo de construção do modelo de dados construído recorrendo apenas a dados de tráfego.

4.3.5 Conjugação dos dados meteorológicos e rodoviários

Para conjugar os dados meteorológicos com os dados de tráfego rodoviário e, consequentemente, criar um conjunto de dados capazes de serem utilizados no modelo de dados desenvolvido elaborou-se um algoritmo em *Python*.

Inicialmente, este algoritmo começa por considerar o último registo meteorológico recebido, extraindo os respetivos valores do ano, mês, dia, hora e minuto. Estes valores serão necessários para identificar o registo correspondente nos dados recolhidos pelos radares.

Posteriormente, o algoritmo itera sobre os registos recolhidos pelos radares de modo a obter o registo corresponde ao mesmo intervalo temporal. Após encontrar este registo, todos

os valores de ambos os tipos de dados são extraídos e guardados numa variável. Finalmente, esta variável é escrita num ficheiro CSV previamente inicializado formando, assim, um ficheiro capaz de ser utilizado pelo modelo de dados desenvolvido. O ficheiro resultante é composto por 24 atributos meteorológicos e 20 atributos relacionados com o tráfego rodoviário existente.

CAPÍTULO 5

Conclusões

5.1 Conclusões sobre o trabalho

Em conclusão, considera-se que a arquitetura proposta cumpre todos os requisitos. No que concerne ao sistema de previsão de tráfego em tempo real em função das condições meteorológicas, observou-se que este cumpre todos os objetivos conjeturados inicialmente.

Apesar de existirem sub-componentes passíveis de serem melhoradas, destacam-se as operações de envio de mensagens e cálculo dos valores necessários referentes aos valores recolhidos. Considera-se que estas operações se encontram totalmente otimizadas, apresentando valores de execução na ordem dos milissegundos (tabela 4.5).

Para além das componentes mencionadas nos parágrafos anteriores, destaca-se também a componente relativa à recolha de dados de tráfego rodoviário. Conclui-se que esta componente satisfaz todas as necessidades propostas nesta dissertação.

Finalmente, percebe-se que o algoritmo de previsão proposto cumpre os requisitos desta dissertação.

Os resultados do algoritmo de previsão escolhido podem ser considerados como aceitáveis em relação ao tempo de execução. Adicionalmente, o erro médio absoluto deste algoritmo pode ser considerado como sendo bom, dado que se encontra entre 8 e 9% e apresenta um desvio padrão baixo (0,48).

Foram realizados testes ao modelo de dados com dois tipos de dados distintos: dados meteorológicos conjugados com dados de tráfego e dados de tráfego (exclusivamente). Concluiu-se que a utilização de dados meteorológicos e de tráfego conjugados era mais vantajosa que a utilização exclusiva de dados de tráfego, dado que os primeiros apresentam menor erro médio absoluto e também menor tempo de execução para a construção do modelo de dados.

Em suma, considera-se que todos os objetivos inicialmente propostos para esta dissertação foram atingidos.

5.2 Limitações e identificação de trabalho futuro

Compreende-se que as componentes da arquitetura podem ser ampliadas, melhorando as suas capacidades. Assim, considera-se que esta expansão poderia ser alcançada a partir de dois modos: aumento do número de ligações a componentes externos (por exemplo, outras plataformas semelhantes ao *Traffic Control Center* [28]) e também aumento dos *endpoints* disponíveis na API.

Conclui-se que componente responsável pela recolha de dados meteorológicos apresenta sub-componentes que podem ser aperfeiçoadas, nomeadamente:

- aumentar o número de fontes de dados meteorológicos (consequentemente, aumentar o volume de dados recolhidos e também o tipo de valores recolhidos);
- conjugar dados de diversas fontes (por exemplo, conjugação de dados entre diferentes estações meteorológicas);
- ampliar o tipo de dados recolhidos (por exemplo, direção do vento);
- estabelecer uma parceria com uma entidade externa (por exemplo, European Centre for Medium-Range Weather Forecasts (ECMWF) ou outra entidade de renome internacional) de modo a obter dados mais precisos e mais frequentes;
- diminuição do tempo necessário para obter o último registo meteorológico tornando, consequentemente, o fluxo de mensagens mais rápido.

Uma análise mais aprofundada evidencia que a componente relativa à recolha de dados de tráfego rodoviário pode ser melhorada, apesar do bom desempenho demonstrado pela mesma. Assim, constata-se que uma das possíveis formas de melhorar esta componente estaria relacionada com o aumento do número de radares responsáveis pela recolha de dados. Para além disso, após o incremento do número de radares, surgem outras questões que requerem reflexão por parte dos responsáveis pela sua colocação, nomeadamente, a disposição dos mesmos. Acredita-se que um maior número de radares disponíveis e, consequentemente, uma melhor disposição iria fornecer mais e melhores dados de tráfego.

De modo a aumentar a precisão das previsões realizadas (isto é, diminuir o erro médio absoluto) poderiam ser testados outros algoritmos de previsão que não foram considerados nesta dissertação, nomeadamente, RNN, RBFNN, MLP, entre outros. Acredita-se que, em comparação com outros algoritmos de previsão, CNN é, atualmente, a melhor opção (considerando precisão e tempo de execução).

Referências

[1] *PORDATA - Veículos matriculados: total e por tipo de veículo*. URL: https://www.pordata.pt/Portugal/Ve%c3%adculos+matriculados+total+e+por+tipo+de+ve%c3%adculo-3103-262503.

[2] *5G-MOBIX*. URL: https://www.5g-mobix.com/.

[3] *IPMA - Serviços*. URL: https://www.ipma.pt/pt/produtoseservicos/index.jsp?page=dados.xml.

[4] J. Ferreira, F. Braz, F. Baldo e P. Gonçalves, «Analyzing the impact of the covid epidemic on beach access traffic,» em *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, 2022, pp. 1–7. DOI: 10.23919/CISTI54924.2022.9820252.

[5] F. J. Braz, J. Ferreira, F. Gonçalves et al., «Road Traffic Forecast Based on Meteorological Information through Deep Learning Methods,» *Sensors*, vol. 22, n.º 12, 2022, ISSN: 1424-8220. DOI: 10.3390/s22124485. URL: https://www.mdpi.com/1424-8220/22/12/4485.

[6] D. Ma, X. Song e P. Li, «Daily Traffic Flow Forecasting through a Contextual Convolutional Recurrent Neural Network Modeling Inter- And Intra-Day Traffic Patterns,» *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 2627–2636, 5 mai. de 2021, ISSN: 15580016. DOI: 10.1109/TITS.2020.2973279.

[7] J. Lee, B. Hong, K. Lee e Y.-J. Jang, «A Prediction Model of Traffic Congestion Using Weather Data,» IEEE, dez. de 2015, pp. 81–88, ISBN: 978-1-5090-0214-6. DOI: 10.1109/DSDIS.2015.96. URL: http://ieeexplore.ieee.org/document/7396485/.

[8] I. Lana, J. D. Ser, M. Velez e E. I. Vlahogianni, *Road Traffic Forecasting: Recent Advances and New Challenges*, 2018. DOI: 10.1109/MITS.2018.2806634.

[9] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi e B. Yin, «Deep Learning on Traffic Prediction: Methods, Analysis and Future Directions,» abr. de 2020. DOI: 10.1109/TITS.2021.3054840. URL: http://arxiv.org/abs/2004.08555%20http://dx.doi.org/10.1109/TITS.2021.3054840.

[10] A. M. Nagy e V. Simon, *Survey on traffic prediction in smart cities*, out. de 2018. DOI: 10.1016/j.pmcj.2018.07.004.

[11] C. Badii, P. Nesi e I. Paoli, «Predicting Available Parking Slots on Critical and Regular Services by Exploiting a Range of Open Data,» *IEEE Access*, vol. 6, pp. 44 059–44 071, ago. de 2018, ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2864157.

[12] B. Deb, S. R. Khan, K. T. Hasan, A. H. Khan e A. Alam, «Tavel Time Prediction using Machine Learning and Weather Impact on Traffic Conditions,» mar. de 2019.

[13] S. Shahriar, A. R. Al-Ali, A. H. Osman, S. Dhou e M. Nijim, «Prediction of EV charging behavior using machine learning,» *IEEE Access*, vol. 9, pp. 111 576–111 586, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3103119.

[14] J. Ferreira, J. Fonseca, D. Gomes et al., *PASMO: an open living lab for cooperative ITS and smart regions*. DOI: 10.1109/ISC2.2017.8090866.

[15] P. M. Santos, J. G. Rodrigues, S. B. Cruz et al., «PortoLivingLab: An IoT-Based Sensing Platform for Smart Cities,» *IEEE Internet of Things Journal*, vol. 5, pp. 523–532, 2 abr. de 2018, ISSN: 23274662. DOI: 10.1109/JIOT.2018.2791522.

[16] G. Dudek, «Neural networks for pattern-based short-term load forecasting: A comparative study,» *Neurocomputing*, vol. 205, pp. 64–74, set. de 2016, ISSN: 18728286. DOI: 10.1016/j.neucom.2016.04.021.

[17] Y. Hou, Z. Deng e H. Cui, «Short-Term Traffic Flow Prediction with Weather Conditions: Based on Deep Learning Algorithms and Data Fusion,» *Complexity*, vol. 2021, 2021, ISSN: 10990526. DOI: 10.1155/2021/6662959.

[18] R. Reddy e G. Shyam, «Analysis through machine learning techniques: A survey,» 2018, pp. 542–546, ISBN: 9781538656570. DOI: 10.1109/ICGCIoT.2018.8753050.

[19] C. Bentéjac, A. Csörgő e G. Martínez-Muñoz, «A Comparative Analysis of XGBoost,» nov. de 2019. DOI: 10.1007/s10462-020-09896-5. URL: http://arxiv.org/abs/1911.01914%20http://dx.doi.org/10.1007/s10462-020-09896-5.

[20] *XGBoost Documentation — xgboost 1.5.1 documentation*. URL: https://xgboost.readthedocs.io/en/stable/.

[21] T. Chen e C. Guestrin, «XGBoost: A Scalable Tree Boosting System,» DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[22] *UCI Machine Learning Repository*. URL: https://archive.ics.uci.edu/ml/index.php.

[23] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi e J. Akinjobi, «Supervised machine learning algorithms: classification and comparison,» *International Journal of Computer Trends and Technology*, vol. 48, 3 2017.

[24] R. Murugan e N. Palanichamy, «Smart city air quality prediction using machine learning,» Institute of Electrical e Electronics Engineers Inc., mai. de 2021, pp. 1048–1054, ISBN: 9781665412728. DOI: 10.1109/ICICCS51141.2021.9432074.

[25] *Aimsun: simulation and AI for intelligent mobility*. URL: https://www.aimsun.com/.

[26] *Road Traffic Simulation Software – AnyLogic Simulation Software*. URL: https://www.anylogic.com/road-traffic/.

[27] J. Novakovic, A. Veljovi, S. Iiic, Z. Papic e M. Tomovic, «Evaluation of Classification Models in Machine Learning,» *Theory and Applications of Mathematics & Computer Science*, vol. 7, 1 2017, ISSN: 2247-6202.

[28] *CCam - Traffic Control Center*. URL: https://ccam.av.it.pt/.

[29] *Django - The web framework for perfectionists with deadlines*. URL: https://www.djangoproject.com/.

[30] *Time series forecasting | TensorFlow Core*. URL: https://www.tensorflow.org/tutorials/structured_data/time_series.