Towards synthetic sensing for smart cities : a machine/deep learning-based approach

Faraz Malik Awan

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2022IPPAS006

Thèse de doctorat

Towards Synthetic Sensing for Smart Cities:A Machine/Deep Learning Approach

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Telecom SudParis

École doctorale n° 626: École doctorale de l'institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Evry, le Juin 30, 2022, par

Faraz Malik Awan

Composition du Jury:

Luigi Atzori Professeur, University of Cagliari (MC-LAB)	Rapporteur/ Président
Joongheon Kim Associate Professor, Korea University (AI and Mobility Lab)	Rapporteur
Lila Boukhatem Associate Professor, Universite Paris-Sud XI (Laboratoire LRI)	Examinateur
Payam Boukhatem Professeur, Imperial College London (Department of Brain Sciences)	Examinateur
Noel Crespi Professeur, Telecom SudParis (DICE Lab)	Directeur de thèse
Roberto Minverva Associate Professor, Telecom SudParis (DICE Lab)	Co-Directeur de thès

Titre : Vers une détection synthétique pour les villes intelligentes : une Approche de Machine/Deep Learning

Mots clés : Smart City, Internet des Objets, Machine Learning, Pollution de l'air, Trafic, Smart Parking

Résumé : Il existe une tendance technique et commerciale à l'exploitation des données disponibles pour améliorer les processus et la compréhension des grands phénomènes complexes. Des technologies telles que l'internet des objets (IoT), l'intelligence artificielle (AI) et l'analyse des données ont largement contribué à l'exploitation des données pour contrôler, gouverner et comprendre les dynamiques au sein de grands environnements, apportant de nouvelles fonctions et applications dans la réalité. La ville intelligente est l'un des exemples de projets réussis qui sont possibles grâce à ces technologies. L'un des principaux objectifs du concept de ville intelligente est de rendre les villes mesurables et contrôlables afin d'offrir un meilleur lieu de vie à leurs habitants. Le but est d'utiliser la numérisation pour fournir des services, apporter l'automatisation et proposer une meilleure planification pour les villes. Pour atteindre cet objectif, l'IdO est déterminant. Il permet de mesurer et de collecter des données qui représentent des phénomènes physiques dans des environnements spécifiques. Ces données sont utilisées pour surveiller et gérer des processus destinés à optimiser l'impact attendu sur l'environnement. Cela étant dit, de nombreuses applications sont possibles si des données de bonne qualité sont détectées et mises à disposition. Ces applications peuvent jouer un rôle très important dans la résolution de nombreux problèmes et défis dans les villes. Par exemple, l'utilisation massive de véhicules pose des problèmes et des défis liés à la circulation dans les villes. Ces problèmes comprennent, sans s'y limiter, le stationnement, la circulation et la pollution atmosphérique. Bien que l'intensité de ces problèmes puisse varier d'une région à l'autre, les zones urbaines, et plus particulièrement les métropoles, semblent être les plus touchées. Avec le déploiement croissant des grands réseaux de capteurs, les villes instrumentent de vastes zones pour collecter des données sur la circulation, l'occupation des parkings et la pollution des parkings et la pollution des atmosphérique, pour n'en citer que quelques-unes. Ce vaste réseau de capteurs génère une énorme quantité de données, qui peuvent s'avérer utiles pour résoudre les problèmes susmentionnés. Inspirés et motivés par cette idée, nous avons travaillé sur l'un des axes de recherche les plus importants de la ville intelligente, à savoir les systèmes de transport intelligents (STI). Les ITS englobent plusieurs domaines, tels que le télépéage, les systèmes de notification des véhicules, l'information sur le trafic, le stationnement intelligent et l'environnement. Cependant, dans cette thèse, nous ciblons deux de ses domaines importants : i) le stationnement intelligent et ii) le trafic routier. Nous avons commencé notre recherche par le cas d'utilisation du stationnement intelligent. En faisant une revue de la littérature, nous avons réalisé que différentes approches de Machine Learning (ML) et de Deep Learning (DL) ont été utilisées pour des solutions de stationnement intelligent. Dans la plupart de ces approches proposées, les zones de stationnement fermées étaient ciblées et différents ensembles de caractéristiques étaient utilisés pour prédire le "taux d'occupation" dans ces zones de stationnement. Cela nous a incités à mener une analyse comparative pour répondre aux questions suivantes : compte tenu du cas d'utilisation de la prédiction du stationnement, comment les modèles ML traditionnels se comportent-ils par rapport aux modèles DL complexes ? Avec des données volumineuses, les modèles ML traditionnels moins complexes peuvent-ils surpasser les modèles DL complexes ? Quelle est la performance de ces modèles pour prédire la disponibilité des places de stationnement individuelles dans la rue plutôt que de prédire le taux d'occupation global d'une zone de stationnement fermée. Pour répondre à ces questions, nous avons choisi trois algorithmes ML classiques bien connus (K-Nearest Neighbours, Random Forest, Decision Tree) pour les comparer à un algorithme DL (Multilayer Perceptron). Afin d'approfondir notre étude, nous formons un modèle

Institut Polytechnique de Paris
91120 Palaiseau, France

d'apprentissage d'ensemble (également connu sous le nom de classificateur de vote), dans lequel nous combinons tous les modèles ML et DL susmentionnés. Dans ce travail, nous utilisons un énorme ensemble de données de stationnement de la ville de Santander, en Espagne, qui se compose d'environ 25 millions d'enregistrements. En outre, nous ciblons les places de stationnement individuelles plutôt que le taux d'occupation d'une zone de stationnement entière. Nous proposons également de recommander des places de stationnement disponibles en fonction de la position actuelle du conducteur. Dans le cadre de nos objectifs de recherche, nous avons effectué une analyse documentaire approfondie du trafic routier, de son influence sur l'environnement et des défis et problèmes qui y sont liés. Dans la littérature, le trafic routier est souvent associé à la pollution atmosphérique et à la pollution sonore. Une forte corrélation entre le trafic routier et la pollution atmosphérique et sonore a été démontrée dans de nombreux travaux disponibles. De plus, dans beaucoup de ces travaux, le trafic routier a été utilisé pour prédire la pollution atmosphérique et la pollution sonore. Cependant, à notre connaissance, la pollution atmosphérique et la pollution sonore n'ont jamais été utilisées dans le problème de la prédiction du trafic. Dans cette partie de notre recherche, nous avons d'abord utilisé la pollution de l'air (CO, NO, NO2, NOx, et O3) avec les variables atmosphériques, telles que la vitesse et la direction du vent, la température et la pression, pour améliorer la prévision du trafic dans la ville de Madrid. Cette expérience réussie nous a incités à étendre notre étude à une autre entité, qui est également fortement corrélée au trafic routier, à savoir la pollution sonore. Ainsi, en tant qu'extension de notre travail précédent, dans cette partie de notre recherche, nous utilisons la pollution sonore pour améliorer la prévision du trafic dans la ville de Madrid.

Title : Towards Synthetic Sensing for Smart Cities: A Machine/Deep Learning Approach

Keywords : Smart City, Internet of Things, Machine Learning, Air Pollution, Traffic, Smart Parking

Abstract : There is a technical and business trend towards the exploitation of available data for improving the processes and the comprehension of large complex phenomena. Technologies such as Internet-of Things (IoT), Artificial Intelligence (AI), and Data Analysis largely contributed to exploit data to control, govern, and understand dynamics within large environments bringing new functions and applications into reality. Smart city is one of the examples of successful projects that are possible because of these technologies. One of the main objectives of the concept of smart cities is to make cities measurable and controllable in order to offer a better place to live for their inhabitants. The goal is to use digitization to provide services, bring automation, and come up with a better planning for the cities. To accomplish this goal, the IoT is instrumental. It enables to measure and collect data that represent physical phenomena in specific environments. These data are used to monitor and govern processes that are meant to optimize the expected impact on the environment. With that being said, plenty of applications are possible if good quality data are sensed and openly made available. Such applications can play a very important role in tackling with many issues and challenges in cities. For example, the massive usage of vehicles is posing the issues and challenges related to traffic in cities. These issues include, but not limited to, parking, traffic flow, and air pollution. Though the intensity of these issues may vary from area to area, urban areas, more particularly metropolitan cities, seem getting effected the most. With the growing deployment of large sensor networks, cities are instrumenting large areas for collecting traffic flow, parking occupancy, and air pollution information/data, to name a few. This large network of sensors generates a huge amount of data, which can come handy in order to tackle with above-mentioned issues. Being inspired and motivated by this idea, we worked on one of the most significant research directions in Smart City, i.e., Intelligent Transportation System (ITS). ITS encapsulates several domains, such as electronic toll collection, vehicles notification systems, traffic information, smart parking, and environment. However, in this thesis, we target two of its

important domains; i) Smart Parking, and ii) Road Traffic. We started our research with Smart Parking use case. While doing literature review, we realized that different Machine Learning (ML) and Deep Learning (DL) approaches have been used for smart parking solutions. In most of these proposed approaches, enclosed parking areas were targeted and different feature sets were used to predict the "occupancy rate" in those parking areas. It inspired us to conduct a comparative analysis to answer following questions; Given the parking prediction use case, how do the traditional ML models perform as compared to complex DL models? Provided big data, can less complex, traditional ML models outperform complex DL models? How well these models can perform to predict the availability of the individual on-street parking spots rather than predicting the overall occupancy rate of an enclosed parking area. To answer these questions, we choose three well-known classical ML algorithms (K-Nearest Neighbours, Random Forest, Decision Tree) to perform comparison with a DL algorithm (Multilayer Perceptron). In order to take our investigation into depth, we train Ensemble Learning Model (also known as Voting Classifier), in which we combine all the above-mentioned ML and DL models. In this work, we use a huge parking dataset of city of Santander, Spain, which consists of around 25 million records. Furthermore, we target individual parking spot rather than the occupancy rate of an entire parking area. We also propose to recommend available parking spots based on the current location of the driver. Moving forward with our research goals, we performed in depth literature review on road traffic, its influence on environment, and challenges and issues related to it. In the literature, road traffic is often associated with air pollution and noise pollution. A strong correlation between road traffic and air & noise pollution has been shown in many works available. Furthermore, in many of these works, road traffic has been used to predict air pollution and noise pollution. However, to the best of our knowledge, air pollution & noise pollution have never been used in traffic prediction problem. In this part of our research, firstly we used air pollution (CO, NO, NO2, NOx, and O3) along with the atmospheric variables, such as wind speed, wind direction, temperature, and pressure to improve the traffic forecasting in the city of Madrid. This successful experiment motivated us to extend our investigation to another entity, which is also strongly correlated with road traffic i.e., noise pollution. Hence, as an extension of our previous work, in this part of our research, we use noise pollution to improve the traffic prediction in the city of Madrid.

Doctor of Philosophy (PhD) Thesis
Institut-Mines Télécom, Télécom SudParis & Institut Polytechnique de Paris (IP Paris)

Specialization

Computer Science - Artificial Intelligence

presented by

Faraz Malik Awan

Towards Synthetic Sensing for Smart Cities: A Machine/Deep Learning Approach

Committee:

Luigi Atzori	Reviewer	Professor, University of Cagliari - Italy
Joongheon Kim	Reviewer	Associate Professor, Korea University - Korea
Payam Barnaghi	Examiner	Professor, Imperial College London- UK
Lila Boukhatem	Examiner	Associate Professor, Universite Paris-Sud XI - France
Noel Crespi	Advisor	Professor, IMT, Telecom SudParis - France
Roberto Minerva	Co-supervisor	Associate Professor, IMT, Telecom SudParis - France

Thèse de Doctorat (PhD) de
Institut-Mines Télécom, Télécom SudParis
et l'Institut Polytechnique de Paris (IP Paris)

Spécialité

Informatique

présentée par

Faraz Malik Awan

Vers une détection synthétique pour les villes intelligentes : une Approche de Machine/Deep Learning

Jury composé de :

Luigi Atzori	Rapporteur	Professor, University of Cagliari - Italy
Joongheon Kim	Rapporteur	Associate Professor, Korea University - Korea
Payam Barnaghi	Examinateur	Professor, Imperial College London- UK
Lila Boukhatem	Examinateur	Associate Professor, Universite Paris-Sud XI - France
Noel Crespi	Directeur de thèse	Professor, IMT, Telecom SudParis - France
Roberto Minerva	Co-directeur de thèse	Associate Professor, IMT, Telecom SudParis - France

Dedication

I dedicate this thesis to my family, specially my father and my mother, who believed in me and supported me in my ups and down

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Noel Crespi, for his continuous support, patience, friendship, insights, as well as all the guidance and help he provided throughout my research. Thank you so much for believing in me throughout this journey and for giving me the freedom to pursue my interests and follow my curiosity. Being a part of your team, is an honor for me.

I would like to thank my co-supervisor, Dr. Roberto Minerva, for his technical support, motivation, and fruitful discussions which we had. Thank you so much for being always available to answer my queries and willing and enthusiastic to assist me in any way at any time, and for providing me advice on every entangled situation.

I would like to extend my sincere thanks to my thesis reviewers, Prof. Luigi Atzori and Assoc. Prof. Joongheon Kim, who patiently read this dissertation and provided invaluable comments and suggestions. A special thanks to Prof. Payam Barnaghi, and Assoc Prof. Lila Boukhatemfor being the part of my jury as examiners for my thesis defense.

During my PhD, I got a chance to work on French national project, FAUCON. I would like to address special thanks for people who were involved in that project and provided me an invaluable experience. I also got a chance to work in collaboration with people from a Romanian Space Agency, Terrasigna. I got to learn a lot in this collaboration. I am thankful to all the people involved in this collaboration who gave me this amazing experience.

My special thanks to all the lovely team members of the Data Intelligence and Communication Engineering Lab at TSP, especially Yasir, Praboda, Marzieh, and Koosha for wonderful times we shared. You were always there with a word of encouragement or listening ear and you provided me all the support and friendship that I needed. Special thanks go to the wonderful administrative staff in TSP, Valerie Mateus and Veronique Guy, who were so helpful and friendly and always helped me in dealing with PhD administrative tasks.

My deep and sincere gratitude to my best friend, Najam, who have been a wonderful, supportive, caring, and generous friend, with whom I shared my happiness and sadness.

My profound love, respect and thank goes to my family whom I owe a great deal. I deeply thank my parents, Malik Alamdar Hussain Awan and Naheed Akhtar Malik, for their unrequited love, unconditional trust, timely encouragement, and endless patience. It was their love that empowered me to break my limits and experience the life freely and fearlessly. I would like to thank my beautiful sisters, Bobby Malik and Sanam Malik, for their for always being a great support. I am also grateful to my brothers, Malik Dilnawaz Awan and Malik Dilfaraz Awan who are like backbone in my life. You have been generous with your love and encouragement despite the long distance between us, and I could not have asked for anything better.

At the end, I would like to thank my nieces, Sania, Arisha, and Angeline, and nephews, Sameer and Sarim for always making me laugh with your mischievous acts. I love you!!!

Faraz Malik Awan
28^{th} March 2022

Abstract

There is a technical and business trend towards the exploitation of available data for improving the processes and the comprehension of large complex phenomena. Technologies such as Internet-of Things (IoT), Artificial Intelligence (AI), and Data Analysis largely contributed to exploit data to control, govern, and understand dynamics within large environments bringing new functions and applications into reality. Smart city is one of the examples of successful projects that are possible because of these technologies. One of the main objectives of the concept of smart cities is to make cities measurable and controllable in order to offer a better place to live for their inhabitants. The goal is to use digitization to provide services, bring automation, and come up with a better planning for the cities. To accomplish this goal, the IoT is instrumental. It enables to measure and collect data that represent physical phenomena in specific environments. These data are used to monitor and govern processes that are meant to optimize the expected impact on the environment. With that being said, plenty of applications are possible if good quality data are sensed and openly made available. Such applications can play a very important role in tackling with many issues and challenges in cities. For example, the massive usage of vehicles is posing the issues and challenges related to traffic in cities. These issues include, but not limited to, parking, traffic flow, and air pollution. Though the intensity of these issues may vary from area to area, urban areas, more particularly metropolitan cities, seem getting effected the most.

With the growing deployment of large sensor networks, cities are instrumenting large areas for collecting traffic flow, parking occupancy, and air pollution information/data, to name a few. This large network of sensors generates a huge amount of data, which can come handy in order to tackle with above-mentioned issues. Being inspired and motivated by this idea, we worked on one of the most significant research directions in Smart City, i.e., Intelligent Transportation System (ITS). ITS encapsulates several domains, such as electronic toll collection, vehicles notification systems, traffic information, smart parking, and environment. However, in this thesis, we target two of its important domains; i) Smart Parking, and ii) Road Traffic. We started our research with Smart Parking use case. While doing literature review, we realized that different Machine Learning (ML) and Deep Learning (DL) approaches have been used for smart parking solutions. In most of these proposed approaches, enclosed parking areas were targeted and different feature sets were used to predict the "occupancy rate" in those parking areas. It inspired us to conduct a comparative analysis to answer following questions; Given the parking prediction use case, how do the traditional ML models perform as compared to complex DL models? Provided big data, can less complex, traditional ML models outperform complex DL models? How well these models can perform to predict the availability of the individual on-street parking spots rather than predicting the overall occupancy rate of an enclosed parking area. To answer these questions, we choose three well-known classical ML algorithms (K-Nearest Neighbours, Random Forest, Decision Tree) to perform comparison with a DL algorithm

(Multilayer Perceptron). In order to take our investigation into depth, we train Ensemble Learning Model (also known as Voting Classifier), in which we combine all the above-mentioned ML and DL models. In this work, we use a huge parking dataset of city of Santander, Spain, which consists of around 25 million records. Furthermore, we target individual parking spot rather than the occupancy rate of an entire parking area. We also propose to recommend available parking spots based on the current location of the driver. Moving forward with our research goals, we performed in depth literature review on road traffic, its influence on environment, and challenges and issues related to it. In the literature, road traffic is often associated with air pollution and noise pollution. A strong correlation between road traffic and air & noise pollution has been shown in many works available. Furthermore, in many of these works, road traffic has been used to predict air pollution and noise pollution. However, to the best of our knowledge, air pollution & noise pollution have never been used in traffic prediction problem. In this part of our research, firstly we used air pollution (CO, NO, NO2, NOx, and O3) along with the atmospheric variables, such as wind speed, wind direction, temperature, and pressure to improve the traffic forecasting in the city of Madrid. This successful experiment motivated us to extend our investigation to another entity, which is also strongly correlated with road traffic i.e., noise pollution. Hence, as an extension of our previous work, in this part of our research, we use noise pollution to improve the traffic prediction in the city of Madrid.

Keywords

Smart City, Internet of Things, IoT, Machine Learning, Deep Learning, Data Analysis, Sensors, Air Pollution, Noise Pollution, Atmospheric Data, Smart Parking, Traffic, LSTM RNN, Decision Tree, Random Forest, KNN, Multilayer Perceptron, Ensemble Learning

Résumé

Il existe une tendance technique et commerciale à l'exploitation des données disponibles pour améliorer les processus et la compréhension des grands phénomènes complexes. Des technologies telles que l'internet des objets (IoT), l'intelligence artificielle (AI) et l'analyse des données ont largement contribué à l'exploitation des données pour contrôler, gouverner et comprendre les dynamiques au sein de grands environnements, apportant de nouvelles fonctions et applications dans la réalité. La ville intelligente est l'un des exemples de projets réussis qui sont possibles grâce à ces technologies. L'un des principaux objectifs du concept de ville intelligente est de rendre les villes mesurables et contrôlables afin d'offrir un meilleur lieu de vie à leurs habitants. Le but est d'utiliser la numérisation pour fournir des services, apporter l'automatisation et proposer une meilleure planification pour les villes. Pour atteindre cet objectif, l'IdO est déterminant. Il permet de mesurer et de collecter des données qui représentent des phénomènes physiques dans des environnements spécifiques. Ces données sont utilisées pour surveiller et gérer des processus destinés à optimiser l'impact attendu sur l'environnement. Cela étant dit, de nombreuses applications sont possibles si des données de bonne qualité sont détectées et mises à disposition. Ces applications peuvent jouer un rôle très important dans la résolution de nombreux problèmes et défis dans les villes. Par exemple, l'utilisation massive de véhicules pose des problèmes et des défis liés à la circulation dans les villes. Ces problèmes comprennent, sans s'y limiter, le stationnement, la circulation et la pollution atmosphérique. Bien que l'intensité de ces problèmes puisse varier d'une région à l'autre, les zones urbaines, et plus particulièrement les métropoles, semblent être les plus touchées. Avec le déploiement croissant des grands réseaux de capteurs, les villes instrumentent de vastes zones pour collecter des données sur la circulation, l'occupation des parkings et la pollution atmosphérique, pour n'en citer que quelques-unes. Ce vaste réseau de capteurs génère une énorme quantité de données, qui peuvent s'avérer utiles pour résoudre les problèmes susmentionnés. Inspirés et motivés par cette idée, nous avons travaillé sur l'un des axes de recherche les plus importants de la ville intelligente, à savoir les systèmes de transport intelligents (STI). Les ITS englobent plusieurs domaines, tels que le télépéage, les systèmes de notification des véhicules, l'information sur le trafic, le stationnement intelligent et l'environnement. Cependant, dans cette thèse, nous ciblons deux de ses domaines importants : i) le stationnement intelligent et ii) le trafic routier. Nous avons commencé notre recherche par le cas d'utilisation du stationnement intelligent. En faisant une revue de la littérature, nous avons réalisé que différentes approches de Machine Learning (ML) et de Deep Learning (DL) ont été utilisées pour des solutions de stationnement intelligent. Dans la plupart de ces approches proposées, les zones de stationnement fermées étaient ciblées et différents ensembles de caractéristiques étaient utilisés pour prédire le "taux d'occupation" dans ces zones de stationnement. Cela nous a incités à mener une analyse comparative pour répondre aux questions suivantes : compte tenu du cas d'utilisation de la prédiction du stationnement, comment les modèles ML traditionnels se comportent-ils par rapport aux modèles DL complexes ? Avec des données volumineuses, les modèles ML

traditionnels moins complexes peuvent-ils surpasser les modèles DL complexes ? Quelle est la performance de ces modèles pour prédire la disponibilité des places de stationnement individuelles dans la rue plutôt que de prédire le taux d'occupation global d'une zone de stationnement fermée. Pour répondre à ces questions, nous avons choisi trois algorithmes ML classiques bien connus (K-Nearest Neighbours, Random Forest, Decision Tree) pour les comparer à un algorithme DL (Multilayer Perceptron). Afin d'approfondir notre étude, nous formons un modèle d'apprentissage d'ensemble (également connu sous le nom de classificateur de vote), dans lequel nous combinons tous les modèles ML et DL susmentionnés. Dans ce travail, nous utilisons un énorme ensemble de données de stationnement de la ville de Santander, en Espagne, qui se compose d'environ 25 millions d'enregistrements. En outre, nous ciblons les places de stationnement individuelles plutôt que le taux d'occupation d'une zone de stationnement entière. Nous proposons également de recommander des places de stationnement disponibles en fonction de la position actuelle du conducteur. Dans le cadre de nos objectifs de recherche, nous avons effectué une analyse documentaire approfondie du trafic routier, de son influence sur l'environnement et des défis et problèmes qui y sont liés. Dans la littérature, le trafic routier est souvent associé à la pollution atmosphérique et à la pollution sonore. Une forte corrélation entre le trafic routier et la pollution atmosphérique et sonore a été démontrée dans de nombreux travaux disponibles. De plus, dans beaucoup de ces travaux, le trafic routier a été utilisé pour prédire la pollution atmosphérique et la pollution sonore. Cependant, à notre connaissance, la pollution atmosphérique et la pollution sonore n'ont jamais été utilisées dans le problème de la prédiction du trafic. Dans cette partie de notre recherche, nous avons d'abord utilisé la pollution de l'air (CO, NO, NO2, NOx, et O3) avec les variables atmosphériques, telles que la vitesse et la direction du vent, la température et la pression, pour améliorer la prévision du trafic dans la ville de Madrid. Cette expérience réussie nous a incités à étendre notre étude à une autre entité, qui est également fortement corrélée au trafic routier, à savoir la pollution sonore. Ainsi, en tant qu'extension de notre travail précédent, dans cette partie de notre recherche, nous utilisons la pollution sonore pour améliorer la prévision du trafic dans la ville de Madrid.

Mots-clés

Smart City, Internet des objets, IoT, Machine Learning, Deep Learning, Analyse de données, Capteurs, Pollution de l'air, Pollution sonore, Données atmosphériques, Smart Parking, Trafic, LSTM RNN, Decision Tree, Random Forest, KNN, Multilayer Perceptron, Apprentissage d'ensemble

Table of contents

Chapter 1

Introduction

Contents

1.1 Motivation

A smart city is defined as the city with the traditional networks and services, which, for the benefit of its inhabitants and businesses, are made more efficient with the use of digital solutions [1]. In other words, smart cities are the cities that improve the management and efficiency of an urban environment with the use of technological solutions. It is currently one of the emerging trends that targets the automation of the monitoring, access, and usage of the infrastructure while supporting the major services offered to the citizens [2]. It also refers to the more interactive and more responsive administration of the city.

Advancement in technologies, such as Internet of Things (IoT), Machine Learning, Data Analysis tools, and 5G Wireless Networks is the fundamental enabler of this concept. In the past few years, Machine Learning, combined with IoT data, played an important role in different domains of smart cities, e.g., mobility, environment, security, and healthcare. As a conventional approach, descriptive and inferential data analysis are performed on the data collected from the IoT and Wireless Sensor Networks (WSNs). Based on the insights from the data analysis, with the help of machine learning, recommendation and/or prediction services are provided for different domains of the smart cities. One of the most successful examples of this process is Intelligent Transportation System (ITS), which deals with smart parking, road traffic information, emergency management etc. Incorporating sensor data and Artificial Intelligence (AI) technologies, main objective of ITS is to provide better information, and safe, reliable, and effective transportation systems to drivers [3] [4]. Transportation systems are one of the most important factors related to economic growth of the countries. Growing number of vehicles are leading to many problems, including accidents, pollution emission, higher fuel prices, and alike [5]. With the availability of latest hardware and software technologies, ITS offers an opportunity to enable better and safer transportation.

Being inspired and motivated by the concept of ITS, in this thesis, we target two of the integral parts of ITS in smart city, i.e., smart parking and road traffic. We chose to work on both domains because they are interrelated in the literature. For example, according to an IBM survey [6], 40% of the traffic in cities is due to the reason that drivers are looking for parking space. Similarly, this relationship exists in other way around too. For example, one of the major concerns of the cities' authorities is that increase in road traffic and congestion may lead to high occupancy rate in the street parking. In many works, road traffic congestion levels have been used to predict parking occupancy by taking into account the relationship between traffic and parking [7] [8]. Similarly, Ziat et al. [9] proposed a joined prediction of road traffic and parking occupancy. Authors used correlation between traffic flow and parking availability to improve the traffic and parking prediction by focusing on

the cross-forecasting of parking availability and road traffic.

As we target two of the important parts of ITS (smart parking and traffic forecasting), the goal of this thesis is twofold; i) comparison of different machine learning/deep learning models for parking prediction system and recommendation system for parking, and ii) proposing approaches to improve road traffic forecasting. More details of the objectives of the thesis are explained in the following section.

1.2 Objectives of the Thesis

We outline the objectives of this thesis in this section. Targeting two of the domains of ITS (smart parking and road traffic), this thesis aims to answer the following questions:

- Given big data of parking, how do different classical machine learning and deep learning models perform for parking space prediction problem? Can classical, less complex machine learning models outperform complex deep learning model?
- Given the relationship between air pollution and road traffic, can the addition of air pollution information in the feature set improve traffic forecasting?
- Like air pollution, noise pollution is also found to be having a correlation with road traffic; can the addition of noise pollution help to improve the road traffic forecasting?

In addition, in the future work of this thesis, we provide the initial directions and roadmap towards the concept of synthetic sensing, which by definition is the usage of one or more type of sensing capabilities to provide the sensing that requires dedicated sensing capabilities.

1.3 Contributions of the Thesis

Our approach to achieve the above mentioned research objectives is organized into three parts as three contributions. We discuss each contribution as follows:

C1: As a first contribution, we analyze and evaluate various ML/DL models and determine the best predictive model among them for the parking space availability problem using the parking space data set of Santander, Spain. For comparison, we present different ML/DL-based solutions, including KNN, Random Forest, Multilayer Perceptron (MLP), Decision Tree, and a combined model called Voting Classifier (or Ensemble Learning). Although there are many ML/DL techniques available in the literature, we chose these five ML/DL techniques because they are, firstly, well-known and widely

used in the community. Secondly, this is a preliminary work which we plan to extend for experimentation and demonstration of the prediction of parking space availability by integrating it into Santander, Spain's smart parking application for validation and to obtain user feedback. We performed this comparison using the well-known evaluation metrics Precision, Recall, F1-Score, and Accuracy. Our contributions are summarized below with respect to the main objective of predicting the availability of parking spaces:

- C1.1 Identification of the best performing, among well-known and generally used ones, AI/ML algorithm, for the problem in hand;
  - C1.1.1 An analysis and evaluation of various ML/DL models (e.g.,KNN, Random Forest, MLP, Decision Tree) for the problem of predicting parking space availability;
  - C1.1.2 An analysis/assessment of the Ensemble Learning approach and its comparison with other ML/DL models; and
  - C1.1.3 Recommendation of the most appropriate ML/DL model to predict parking space availability.
- C1.2 Recommending top-k parking spots with respect to distance between the current position of vehicle and available parking spots;
- C1.3 Application of the algorithms in order to demonstrate how satisfactory prediction of availability of parking spaces can be achieved using real data from Santander;

C2: The second contribution of this thesis is about improving the prediction of traffic intensity in the city of Madrid, using air pollutants and atmospheric data. Details of this contribution are provided below:

- C2.1 We provide a detailed statistical analysis based on the relationship between air pollutants, atmospheric variables, and road traffic;
- C2.2 To the best of our knowledge, this is the first attempt to use air pollutants in combination with atmospheric variables to improve traffic forecasting in a smart city;
- C2.3 Our approach uses a well-known LSTM RNN for time-series traffic data forecasting; and
- C2.4 We provide some proof of the validity of our approach and avenues for future work.

C3: The third contribution of this thesis is an extension work of our second contribution. In this contribution, we investigate the relationship between noise pollution and traffic

intensity on the road. For this purpose, a correlation analysis is conducted. After getting the insights from the data analysis, we use noise pollution to improve the traffic intensity in the city of Madrid. Following the previous contribution, an LSTM Recurrent Neural Network is trained. To evaluate the performance of proposed approach, it is compared with a baseline method which is based on temporal traffic intensity only.

1.4 Publications List

Journal Papers

- F.M. Awan, R. Minerva, N. Crespi, "Using Noise Pollution Data for Traffic Prediction in Smart Cities: Experiments Based on LSTM Recurrent Neural Networks", IEEE Sensors, 2021. [IF=3.301]
- R. Minerva, F.M. Awan, N. Crespi, Exploiting Digital Twin as enablers for Synthetic Sensing, IEEE Internet Computing (Forthcoming), 2021. [IF=4.231]
- F.M. Awan, R. Minerva, and N. Crespi, "Improving Road Traffic Forecasting Using Air Pollution and Atmospheric Data: Experiments based on LSTM Recurrent Neural Networks", MDPI Sensors, 2020. [IF= 3.275]
- F.M. Awan, Y. Saleem, R. Minerva, and N. Crespi, "A Comparative Analysis of Machine/Deep Learning Models for Parking Space Availability Prediction", MDPI Sensors, 2020. [IF= 3.031]

Manuscripts in Progress

- F. M. Awan, Y. Saleem, R. Minerva, N. Crespi, "Urban traffic issues: approaches, methods, tools, challenges, and future perspectives", MDPI Sensors (Prospective Journal)
- F. M. Awan, R. Minerva, N. Crespi, "Major Contributors of Air Pollution in Madrid during and before the COVID period: a statistical analysis", Sustainability (Prospective Journal)

1.5 Relationship of Publications with Contributions

In this section, we provide the relationships of publications with contributions.

- The publication 'A Comparative Analysis of Machine/Deep Learning Models for Parking Space Availability Prediction' corresponds to Contribution C1 in Chapter2.

- The publications 'Improving Road Traffic Forecasting Using Air Pollution and Atmospheric Data: Experiments Based on LSTM Recurrent Neural Networks' corresponds to the contribution C2 in Chapter 3.
- The publication 'Using Noise Pollution Data for Traffic Prediction in Smart Cities: Experiments Based on LSTM Recurrent Neural Networks' corresponds to the contribution C3 in Chapter 4.

1.6 Outline of the Thesis

This thesis consists of 4 chapters.

- Chapter 1: In this chapter, we describe our motivation behind this work, our contributions, publications and their association with the contribution, and the outline of the thesis.
- Chapter 2: This chapter is linked to our first contribution, in which we perform comparative analysis of different machine and deep learning approaches for individual parking spot prediction and recommendation.
- Chapter 3: Our second contribution is associated to this chapter. In this chapter, we describe our approach of improving traffic forecasting using air pollution and atmospheric data.
- Chapter 4: This chapter brackets our third contribution, which is an extension of our work for second contribution. In this chapter, we provide details about the approach of using noise pollution to improve traffic forecasting.
- Chapter 5: Finally, this chapter concludes the thesis and sheds light on the future work directions.

  Related work corresponding to each contribution is provided separately in chapters 2, 3, and 4.

1.7 Ethical Considerations

Regarding General Data Protection Regulation (GDPR) compliance, to respect privacy and ethical aspects, no data containing sensitive and personal information have been collected. Parking data of the city of Santander have been collected as a part of H2020 project, titled WISE-IoT. Whereas the traffic, air pollution, noise pollution, and atmospheric data were collected from open data portal, provided and maintained by Madrid City Council.

Chapter 2

Parking Space Prediction Using Classical ML and Deep Learning Models

Contents

2.1 Overview

Machine/Deep Learning (ML/DL) techniques have been applied to large data sets in order to extract relevant information and for making predictions. The performance and the outcomes of different ML/DL algorithms may vary depending upon the data sets being used, as well as on the suitability of algorithms to the data and the application domain under consideration. Hence, determining which ML/DL algorithm is most suitable for a specific application domain and its related data sets would be a key advantage. To respond to this need, a comparative analysis of well-known ML/DL techniques, including Multilayer Perceptron, K-Nearest Neighbors, Decision Tree, Random Forest, and Voting Classifier (or the Ensemble Learning Approach) for the prediction of parking space availability has been conducted. This comparison utilized Santander's parking data set, initiated while working on the H2020 WISE-IoT project. The data set was used in order to evaluate the considered algorithms and to determine the one offering the best prediction. The results of this analysis show that, regardless of the data set size, the less complex algorithms like Decision Tree, Random Forest, and KNN outperform complex algorithms such as Multilayer Perceptron, in terms of higher prediction accuracy, while providing comparable information for the prediction of parking space availability. In addition, in this chapter, we are providing Top-K parking space recommendations on the basis of distance between current position of vehicles and free parking spots.

2.2 Introduction

2.2.1 Background

One of the most challenging tasks associated with metropolitan cities like Paris or New York or even smaller ones like Santander, Spain is to find an available parking space. According to an IBM survey [6], about 40% of the road traffic in cities is actually composed of vehicles whose drivers are searching for parking spaces. This problem exacerbates issues such as fuel consumption, pollution emission, road congestion, and wasted time, not to mention contributing to accidents due to the drivers' main focus on finding their space [10]. Much work has been done on parking space management, e.g., utilizing sensors (for determining available parking spots) [11] and user feedback (i.e., people informing others of parking space availability by means of applications) to identify available parking spaces [12]. Such systems are based on transient data, without the possibility to actually reserve and allocate the parking spots, and so these techniques are only practical in very short time frames and when the user is in close proximity to the parking areas. Even so, they do not offer any guarantee that a parking spot will be available. However, to predict the availability of free parking

spots at a particular time in the future, these systems coupled with Artificial Intelligence (AI)-based approaches can provide solutions. In order to succeed in the task of predicting parking space availability, data generated by the IoT sensors and the IoT devices, combined with ML/DL approaches, can be very useful. Given the variety of ML/DL methods, one technical problem is to identify the most suitable ML/DL model for the problem and the data set, as the performance of each ML/DL model varies from problem to problem and data set to data set. It is important to mention here some of the relevant works that have been done on comparing AI/ML algorithms in several application domains. The use of ML/DL algorithms has been compared for different application fields. For example, Hazar et al. [13] analyze automatic modulation recognition over Rayleigh fading channels. They trained various ML/DL models for this task, including Random Forest, KNN, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Naïve Bayes, Gradient Boosted Regression Tree (GBRT), Hoeffding Tree, and Logistic regression, and found Naïve Bayes to be an optimal algorithm for this problem. While they ranked GBRT and Logistic Regression as the best algorithms in terms of recognition performance, these algorithms required more processing time. Similarly, Naryanan et al. [14] applied Artificial Neural Network (ANN), KNN, and Support Vector Machine (SVM) approaches to a malware classification problem, and found that KNN outperformed SVM and ANN in terms of accuracy.

2.2.2 Impact of our Parking Prediction Model on Smart Cities

Smart Cities is a widely used term and is an umbrella that accommodates various aspects related to urban research. Mobility and Transportation are considered as the most important branches of the research related to smart cities. Smart transportation and mobility have the potential to make significant contributions in smart cities by utilizing the Internet of Things (IoT) technologies. As described earlier, drivers in search of parking space cause the traffic congestion, affecting many operations and domains of smart cities such as route planning, traffic management, and parking spaces management. Here, the smart parking system makes an effort to reduce the traffic congestion on the roads [15] enriched by our presented parking prediction ML/DL models that makes a significant impact on smart cities. Additionally, since our presented parking prediction models work on the data set of a smart city, Santander, therefore, it can have a direct impact on Santander smart city.

The organization of this paper is as follows. Section 2.3 presents the State of the Art. Section 2.4 provides an overview of the five ML/DL techniques used for our analysis and the performance of these ML/DL techniques is presented in section 2.5.

2.3 Related Work

Many systems have been proposed to deal with the parking spot recommendation problem. The most common solution to this problem is a recommendation system based on real-time sensors capable of detecting parking space availability [11]. For example, Yang et al. [16] evaluated a real-time Wireless Sensor Network (WSN) linked with a web server that collects the data for determining the available parking spots. These data are then passed on to users by means of a mobile phone application. Similarly, Barone et al. [15] proposed an architecture, named Intelligent Parking Assistant (IPA). The proposed architecture does not provide parking spot availability prediction. In fact, it enables users to reserve a parking spot. In order to reserve a parking spot, the user is supposed to get registered with IPA; only the authorized user can use this architecture. Dong et al. [17] present a simulation-based method, Parking Rank, to deal with the real-time detection of parking spots. Their system collected the public information of parking spots, e.g., price, total available space, rented space, etc. and sorted the parking spots by following the Page Rank algorithm. Since they are based on checking real-time data, these systems do not offer the possibility to predict the availability of a parking space in an area and in a time frame (e.g., between 20 and 30 min from the current time) of interest of the user. Therefore, other solutions have been suggested. A Neural Network based model (MLP) was proposed by Vlahogianni et al. [18] to predict the occupancy rate of parking areas and parking spots. For example, in a specific parking area, there is a 75% probability that a parking space is going to be available in 5 min. Badii et al. [19] performed a comparative analysis of Bayesian Regularized Neural Network, Support Vector Regression, Recurrent Neural Network, and Auto-regressive integrated moving average methods for the prediction of parking spot availability within a specific garage without specifying a particular parking spot. With ML/DL models, there are two different research directions: off-street parking spots and on-street parking spots [19]. Their approach is limited to parking spots inside garages with gates (e.g., off-street parking spots). In addition, they included complex features like weather forecasts in their data set. Zheng et al. [20] performed a comparative analysis of Regression Tree, Neural Network, and Support Vector Regression (SVR) methods for the prediction of parking occupancy rates. Since they were dealing with the occupancy rate, while collecting the data they focused on information such as the number of occupied parking spaces. In terms of predicting the parking occupancy rate, Zheng et al. found that the Regression Tree method outperforms the other two algorithms they evaluated. Camero et al. [21] presented a Recurrent Neural Network (RNN)-based approach to predict the number of free parking spaces. Their main aim was to improve the performance of the RNN. To do so, they introduced a Genetic Algorithm (GA)-based technique and searched for the best configuration for RNN using

the GA approach. They utilized the parking data of Birmingham, U.K., which contains the parking occupancy rate for each parking area given the time and date. Yu et al. [22] selected the Auto Regressive Integrated Moving Average (ARIMA) model to predict the number of berths available. ARIMA model is used for making time series forecast. Their experiment was based on a central mall's underground parking and they collected one month data (October 2010). As this is one month data, we believe it might not give clear insight as the parking occupancy pattern can vary in different months. We believe that different factors like public holidays or other kinds of holidays can affect the performance, so one month data might not be enough to have a clear view. Bibi et al. [23] performed car identification in a parking spot. They collected the video from the camera and divided the parking spots into blocks. Their main contribution is to identify any parking spot it occupied or not using image processing. This processing is being done in real-time and does not provide any future prediction. However, their approach can be used for data collection. Similarly, Tătulea et al. [24] detected the parking spaces and identified if the parking spots are occupied or available using computer vision techniques and the camera as a sensor. In order to do that, they performed different steps, including Frame Pre-processing, Adaptive Background Subtraction, Metrics & Measurements, History Creation, Results Merging for Final Classification, and Parking Space Status. Again, this work is not about the future prediction of parking spots.

In contrast to the above-mentioned works, we deal with the prediction of on-street parking in Santander, a smart city of Spain and our prediction models are based on less complex data features. Moreover, we are targeting individual parking spot's occupancy status and can make future prediction about such spots with a validity period of 10 to 20 min. Our prediction has a 10 and 20 min validity because, according to our analysis, during peak hours, parking spots near places like city centers or shopping malls usually do not have the same status (free or occupied) for a longer time interval. Their status changes frequently with 10 to 20 min intervals.

2.4 Overview of ML/DL Techniques

Here, we provide an overview of the ML/DP techniques used to evaluate and analyze a data set in order to predict parking space availability. We compared the MLP, KNN, Decision Tree & Random Forest, and Ensemble Learning/Voting Classifier techniques.

2.4.1 Multilayer Perceptron (MLP) Neural Network

MLP is one of the most well-known types of neural networks. It consists of an input layer, one or more hidden layer(s), and an output layer. Each hidden layer consists of multiple

hidden units (also called neurons or hidden nodes). The value of any hidden unit n in any hidden layer is calculated using Equation (1) [25]:

$$h_n = a(\sum_{K=1}^{N} i_K * W_{K,n}), \tag{2.1}$$

where h_n represents the output value of any hidden unit n in any hidden layer, and a represents the activation function. The activation function is responsible for making the decision related to the activation of a specific hidden unit. N in Equation (1) represents the total number of input nodes (in our case, there are five nodes in the input layer as well as in each of the three hidden layers), and i_K represents the value of input node K being fed to hidden unit h_n. This input node can be an input layer node or it can be a node in any previous hidden layer. $W_{K,n}$ represents the weight of unit h_n. This weight is a measure of the connection strength between an input node and a hidden unit [26].

We used a Rectifier Linear Unit (ReLU) as the activation function for all the layers, so, at each hidden unit, the activation function a in Equation (1), takes the input and returns the output value as follows:

$$O_n = max(0, INP_n), \tag{2.2}$$

where O_n represents the output value of any hidden unit in any hidden layer, $INP_n = \sum_{K=1}^{N} i_K * W_{K,n}$ is the input value of any hidden unit in any hidden layer. ReLU function was recommended as an activation function by the grid search approach (Explained in Section 2.5). Vanishing gradient is one of the major problems faced by DL approaches. Activation functions like Sigmoid and Tanh are not capable of dealing with vanishing gradient problems. However, ReLU does have the ability to deal with vanishing gradient problems [27]. Figure 2.1 illustrates the concept of a fully connected MLP with three hidden layers and with a number of hidden units equal to the number of features $(x_1, x_2, \ldots, x_n)$ in each sample in the data set. The complete details of these features are provided in Section 2.5.

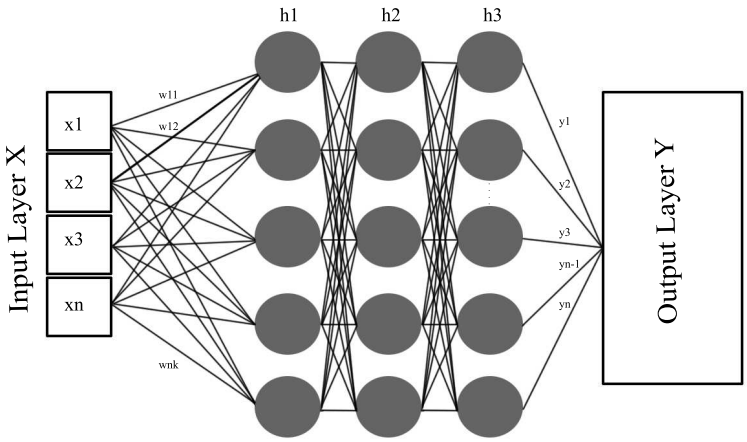


Figure 2.1: MLP architecture.

2.4.2 K-Nearest Neighbors (KNN)

KNN is known as one of the simplest ML algorithms. It classifies samples on the basis of the distances between them. In any classification data set, there are observations in the form of X and Y in the training data, where X_i is the vector containing the feature values, and Y_i is the class label against X_i. Let us suppose there is an observation X_k and we want to predict its class label Y_k using KNN. Still using Equation (3), the KNN algorithm finds the K number of observations in X that are close (or similar) to the observation X_k:

$$DIST_{X_k,X_i} = D(X_k, X_i)_{1 \leq i \leq n}. \tag{2.3}$$

Using Equation (3), the distance between observation X_k and all the observations in X can be calculated. After calculating these distances, the top-K closest (similar) observations from the training data are selected and then classed as the majority among the top-K closest observations is assigned to unlabeled sample X_k. There are several distance functions available, including Manhattan, Minkowski, and Euclidean [28]. Euclidean is the most popular; it calculates the distance between observations using Equation (4):

$$D(X_k, X_i) = \sqrt{\sum_{l=1}^{\#features} (X_{l,k} - X_{l,i}),} \tag{2.4}$$

where $X_{l,k}$ represents the l^{th} feature of sample X_k, and$X_{l,i}$ represents the l^{th} feature of observation X_i.

2.4.3 Decision Tree and Random Forest

The Decision Tree algorithm constructs a tree by setting different conditions on its branches. An exemplary architecture of a Decision Tree is shown in Figure 2.2. It consists of (i) a

root node (i.e., the starting point), (ii) internal nodes (where splitting takes place), and (iii) leaves (Terminal or Final Nodes that contain the homogeneous classes). Again considering the same scenario, with X as the training data set, N as the total number of observations and their corresponding class labels (C) in X, the entropy can be calculated using Equation (5) [29]:

$$E(X) = -\sum_{j-1}^{K} \frac{freq(C_j, X)}{N} \log_2 \frac{freq(C_j, X)}{N}, \tag{2.5}$$

where $\frac{freq(C_j,X)}{N}$ represents class C_j's occurrence probability in X, and N represents the total number of samples in the training set. The information gain is then used to perform node split using equations given in [29].

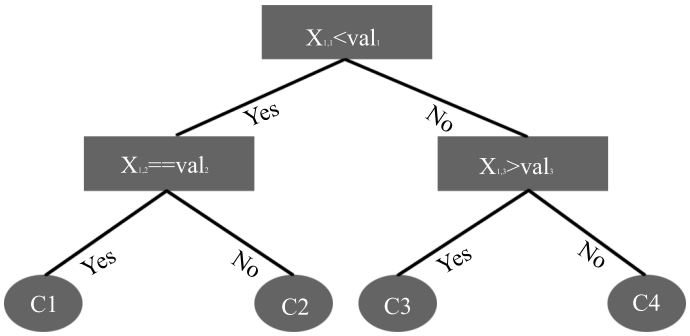


Figure 2.2: Decision tree architecture.

The Random Forest algorithm is similar to the Decision Tree algorithm. In fact, it consists of multiple independent Decision Trees. Each tree in a Random Forest sets conditional features differently. When a sample arrives at a root node, it is forwarded to all the sub-trees. Each sub-tree predicts the class label for that particular sample. At the end, the class in the majority is assigned to that sample.

2.4.4 Ensemble Learning Approach (Voting Classifier)

Figure 2.3 illustrates the concept of Voting Classifier, also known as the Ensemble Learning Approach that combines multiple ML/DL models. In this chapter, we combined MLP, KNN, Decision Tree, and Random Forest algorithms to solve the problem of predicting the availability of parking spaces. The Ensemble Learning approach takes the training data and trains each model. After the training process, the Ensemble Learning approach feeds the testing data to the models and then each model predicts a class label for each sample in the testing data. In the next stage, a voting process is performed for each sample prediction.

Generally, two kinds of voting are available: hard voting and soft voting. In *hard voting*, the Ensemble Learning approach assigns a class label, voted by majority, to the sample. For example, among five models, three models identify that the same sample X_k belongs to Class C_1 while the other two models identify that this sample belongs to Class C_2. Given that Class C_1 has been voted for by the majority, Class C_1 would be assigned to that particular sample.

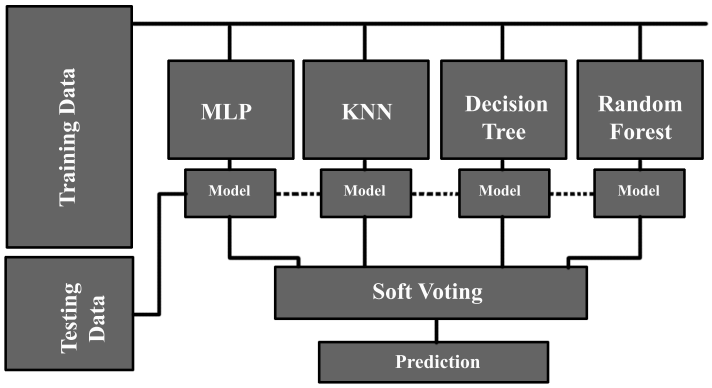


Figure 2.3: Ensemble Learning or Voting Classifier Architecture.

Soft voting, on the other hand, averages the probability of all the expected outputs, i.e., the class labels, and then the class with the highest probability is assigned to the sample.

2.5 Results and Evaluation

During this work, the algorithms described in the previous section have been used, fine-tuned, analyzed and compared with respect to the specific goal of the recommendation system: i.e., to suggest drivers the most probable and closest location to their final destination for a free parking space by looking ahead in a specific time frame (e.g., 20 min times frame). Data were collected by sensors deployed in a real environment, i.e., the smart city Santander. In this section, we evaluate the performance of five ML/DL models for the prediction of parking space availability and provide a comparative analysis of the preliminary results, which we plan to extend by integrating them into a smart parking application for Santander, Spain for future experimentation.

2.5.1 Parking Space Data Set

The data set for the prediction was obtained by collecting the measurements of sensors deployed in Santander, a smart city in Spain. Almost 400 on-street parking sensors are

deployed in the main parking areas of the city center. These parking sensors [30] capture the status (i.e., occupied or free) of the parking spots. Collected over a 9-month period, this data set was constructed as part of the WISE-IoT [31], an H2020 EU-KR project. In WISE-IoT, the parking sensor data was stored in an Next Generation Service Interface (NGSI) context broker [32]. We accessed real-time Santander data; in the WISE-IoT project, in order to make data more consistent, we created a script that retrieves and stores the on-street parking sensor data every minute. The objective is twofold: to predict the parking spot availability within a time interval (validity) of 10 to 20 min, and to evaluate the prediction accuracy. The collected data set has around 25 million records. We conducted our initial experiment using data set having around 3 million records. Later on, in order to check the impact of larger data set on the algorithms, the data set was extended to 25 million records. As scaling up the data set size did not affect the standing (ranking) of ML/DL algorithms, we present the results for 25 million records in the Performance Evaluation section. The collected data set has the following organization:

- **Parking ID:** Refers to the unique ID associated with each parking space.
- **Timestamp:** The Timestamp of the parking space data collection.
- **Start Time/End Time:** Start Time and End Time refer to the time interval during which a parking space's status remained the same, i.e., available or occupied.
- **Duration:** Refers to the total duration in seconds during which a specific parking space remained available or remained occupied.
- **Status:** This feature represents the status of a parking space, e.g., available or occupied.

The above-mentioned features were further organized to be input features for our ML/DL model, as given in Table 2.1.

Table 2.1: Extracted features.

Features	**Value/Range**
Parking Spot ID	Unique ID of Sensor
Date	1–30/31 (Date of the month)
Day	1–7 (Day of the week)
Start Hour	0–23
Start Minute	0–59
End Hour	0–23
End Minute	0–59
Status	0–1 (Occupied or Free)

Start hour, start minute and end hour, end minute in Table 2.1 present the 10 or 20 min interval status for any particular parking spot. We collected our data set after every minute; therefore, in order to get the 10 and 20-min status and to provide predictions with 10 and 20 min validity, we used 10 and 20-min windows with 60% and 80% thresholds. For example, if specific parking spot had a 60% availability rate, then, for that 10 or 20-min window, the status of that particular parking spot would be considered available (Free). Similarly, for an 80% threshold, a parking spot would need to have an 80% availability rate in a 10- or 20-min window to be classified "Free".

2.5.2 Hyper-Parameters of ML/DL Techniques

Table 2.2 presents the hyper-parameters of the five ML/DL models that we tuned for our comparative analysis. We used GridSearch [33] in order to get the best hyper-parameters' values for each Machine/Deep Learning model. For MLP, we tuned four hyper-parameters. "Activation" is responsible for determining how active a specific neuron (hidden unit) is. We adopted the widely-used ReLU activation function. As shown in Equation (2), it returns either 0 or the input itself, and then selects the maximum value between 0 and the input value. This means that, if the input value of a neuron is negative, it will return 0 to keep the output of a neuron within range [0, input value]. "hidden_layer_sizes" defines the number of hidden layers and the number of neurons in each hidden layer.

Table 2.2: Hyper-parameters of ML/DL techniques.

MLP		KNN		Decision Tree		Random Forest		Voting Classifier	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
activation	ReLU	n_neighbors	11	max_depth	100	max_depth	100	estimators	MLP, KNN, Random Forest, Decision Tree
early_stopping	True	metric	euclidean	criterion	entropy	criterion	entropy	voting	soft
hidden_layer_sizes	(5,5,5)	n-jobs	None	min_samples_leaf	5	min_samples_leaf	1	weights	1,1,1,2
learning_rate	Adaptive	weights	uniform			n_estimators	200		
learning_rate_init	0.001								
solver	sgd								
tol	0.0001								

ML=Machine Learning, DL=Deep Learning, MLP=Multilayer Perceptron, KNN=K-Nearest Neighbors

In our case, its value is (5,5,5), which shows that three hidden layers with five neurons in each layer are being used in the network. We used some rules of thumb [34] to determine the range of the hidden layer sizes and the neuron sizes. The hyper-parameters "learning_rate" and "learning_rate_init" are responsible for the optimization and minimization of the loss function. We used the "adaptive" learning rate. When the learning rate is set to "adaptive", it keeps the learning rate constantly equal to the initial learning rate as long as there is a decrease in the training loss in each epoch. Every time two consecutive epochs fail to show a decrease in loss function by at least "tol" (tolerance, a float variable for optimization, we used its value = 0.0001), the learning rate is divided by 5. Similarly, for KNN, we tuned four hyper-parameters (i) n_neighbors; (ii) distance metric (Euclidean); (iii) n_jobs (Parallel jobs in search of the nearest neighbors); and (iv) weights (when this is set to uniform all neighboring points are weighted equally). Initially, we did experiments with different numbers of neighbors (1, 5, 7, 11, 25, 50 and 100). "n_neighbors = 11" proved the best option. (Later on, GridSearch also suggested 11 as an optimized parameter). We tuned three hyper-parameters for a Decision Tree. "max_depth" defines the maximum depth of the tree. When it is set to "None", nodes keep expanding until all the leaves end up having only one class in them, or until all the leaves have samples less than min_samples_split in them. However, having a Decision Tree that is too deep could lead to the problem of overfitting. "min_samples_split" represents the minimum number of samples required for a node to go for a further split. Similarly, "min_samples_leaf" defines how many samples a leaf node can contain. "criterion = entropy" works on information gain, which is the information related to the decrease in entropy after a split. "n_estimators" defines the number of trees in the forest. Its default value is 10. As we have a huge data set (∼25 million records), we keep the number of estimators close to the usually-recommended range for a huge data set (i.e., 128 to 200). For an Ensemble Learning approach, the hyper-parameter "estimators" defines

the ML/DL models to be used for prediction, while the hyper-parameter "weights" defines the priority given to each estimator. We assigned equal weights to all the estimators except Decision Tree. We gave Decision Tree a higher priority, as it performed relatively better than the rest of the ML/DL models when it was used alone for parking space prediction. The hyper-parameter "voting" is described in Section 2.4.

2.5.3 Evaluation Metrics

The performance metrics we used for the evaluation and comparison of ML/DL models are given below. Moreover, to check the overfitting and stability of these models, we performed K-fold cross-validation. Each evaluation metric and K-fold cross-validation are explained below:

- *Precision* can be defined as the fraction of all the samples labelled as positive and that are actually positive [35]. It can be mathematically presented as follows:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}. \tag{2.6}$$

- *Recall*, in contrast, is defined as the fraction of all the positive samples; they are also labeled as positive [35]. Mathematical presentation of recall is given below:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}. \tag{2.7}$$

- The *F1-Score* is defined as the harmonic mean of recall and precision [35], defined mathematically as:

$$F1 - Score = \frac{2 * (Recall * Precision)}{Recall + Precision}. \tag{2.8}$$

- *Accuracy* is the measure of the correctly predicted samples among all the samples, expressed in an equation as:

$$Accuracy = \frac{\#CorrectPredictions}{\#TotalSamples}. \tag{2.9}$$

- *K-fold cross-validation* is a method for checking the overfitting and evaluating how consistent a specific model is. In K-fold validation, a data set is divided into K equal sets. Among those K sets, each set is used once as testing data and the remaining sets are used as training data. In this chapter, we used 5-fold cross-validation.

2.5.4 Performance Evaluation

This section provides an evaluation of the performance of the MLP, KNN, decision tree, random forest, and Ensemble Learning algorithms in terms of scores related to each cross-validation. A comparative analysis for 10-min and 20-min prediction was done, considering 60% and 80% thresholds for both predictions.

2.5.4.1 10-Min Prediction Validity (60% Threshold)

Table 2.3 presents the average cross-validation score of MLP, KNN, Random Forest, Decision Tree, and Ensemble Learning models given 10-min predictions with a 60% threshold. It can be seen that the computationally complex model, MLP, showed the lowest performance with an average of 64.63% precision, 52.09% recall, 57.68% F1-Score, and 70.48% accuracy. In contrast, one of the simplest ML models, KNN, outperformed MLP with the results of 73.04% precision, 67.46% recall, 70.14% F1-Score, and 76.71% accuracy. Random Forest performed even better, with 86.90%, 80.11%, 83.37%, and 86.50% for average precision, recall, F1-Score, and accuracy, respectively. Decision Tree's and Ensemble Learning's performances were quite close to each other. Decision Tree showed 91.12% average precision while Ensemble learning had 92.79% average precision. The average recall scores for Decision Tree and Ensemble Learning were 90.28% and 89.24%, respectively. The average F1-Score for Decision Tree was 90.69% while Ensemble Learning showed 90.98%. The average accuracy for Decision Tree was 92.25%, while Ensemble Learning, despite combining all the models, could achieve 92.54% accuracy, an improvement of only 0.29%.

Table 2.3: Average cross validation score of each model (10-min prediction validity with a 60% threshold).

Metrics	MLP	KNN	RF	DT	EL
Precision	64.63	73.04	86.90	91.12	92.79
Recall	52.09	67.46	80.11	90.28	89.24
F1-Score	57.68	70.14	83.37	90.69	90.98
Accuracy	70.48	76.71	86.50	92.25	92.54

MLP=Multilayer Perceptron, KNN=K-Nearest Neighbors,
RF= Random Forest, DT=Decision Tree, EL=Ensemble Learning

2.5.4.2 10-Min Prediction Validity (80% Threshold)

Table 2.4 presents the average cross-validation scores of the ML/DL models given a 10-min prediction validity with an 80% threshold. Following the 60% threshold trend, MLP

performed the worst among all the models being compared. MLP showed 70.48% average accuracy with 64.63% average precision, 52.09% average recall, and 57.68% average F1-Score. KNN had a 76.71% average accuracy, 73.04% average precision, 67.46% average recall, and a 70.71% average F1-Score. Random Forest's average accuracy was 86.50% while its average precision, recall, and F1-Score were 86.90%, 80.11%, and 83.37%, respectively. Again, Decision Tree and Ensemble Learning showed quite similar performances, both at the top end. The average accuracy for Decision Tree and Ensemble Learning was 92.39% and 92.60%, respectively. The average precision shown by Decision Tree was 91.11%, while it was 93.01% for Ensemble Learning. Recall and F1-Score for Decision Tree were 90.32% and 90.71%, respectively. For Ensemble learning, average recall was 88.87% and average F1-Score was 90.89%.

Table 2.4: Average cross validation score of each model (10-min prediction validity with 80% threshold).

Metrics	MLP	KNN	RF	DT	EL
Precision	63.92	73.19	87.01	91.11	93.01
Recall	51.64	67.23	79.86	90.32	88.87
F1-Score	57.13	70.08	83.28	90.71	90.89
Accuracy	71.14	77.18	86.70	92.39	92.60

MLP=Multilayer Perceptron, KNN=K-Nearest Neighbors,
RF= Random Forest, DT=Decision Tree, EL=Ensemble Learning

2.5.4.3 20-Min Prediction Validity (60% Threshold)

In this section, we present the comparative analysis given a 20-min predication validity with a 60% threshold. Table 2.5 presents the average cross-validation score for each model. MLP, the lowest scorer overall, showed 64.97% and 52.16% for precision and recall, respectively. With the F1-Score being dependent on precision and recall, MLP's remained low at 57.83%. MLP's average accuracy was 70.83%. The performance of KNN remained better than that of MLP. It showed 74.15% in average precision, 68.76% for average recall, 71.35% as its average F1-Score, and 77.71% for average accuracy. Random Forest again performed better than these first two, with 82.44% in average precision, 73.78% for average recall, 77.87% as its average F1-Score, and 82.49% for average accuracy. Decision Tree and Ensemble Learning, following their earlier trend, gave very similar performances. Average accuracy for Decision Tree and Ensemble Learning was 87.66% and 88.73%, respectively. The average precision and average recall shown by Decision Tree were 85.64% and 84.37%, respectively, while these were 88.65% and 83.56% for Ensemble Learning. The F1-Scores for Decision

Tree and Ensemble Learning were 85% and 86.03%, respectively.

Table 2.5: Average cross validation score of each model (20-min prediction validity with a 60% threshold).

Metrics	MLP	KNN	RF	DT	EL
Precision	64.87	74.15	82.44	85.64	88.65
Recall	52.16	68.76	73.78	84.37	83.56
F1-Score	57.83	71.35	77.87	85.00	86.03
Accuracy	70.83	77.71	82.49	87.66	88.73

MLP=Multilayer Perceptron, KNN=K-Nearest Neighbors,
RF= Random Forest, DT=Decision Tree, EL=Ensemble Learning

2.5.4.4 20-Min Prediction Validity (80% Threshold)

Here, we present the evaluation results of all the ML/DL models given a 20-min prediction validity and an 80% threshold.

Tables 2.6 shows that, as with the previous experiments, the threshold value did not affect the standing of ML/DL models for this configuration (prediction validity of 20 min and an 80% threshold). Decision Tree and Ensemble Learning remained the top two performers in terms of all evaluation metrics. Ensemble Learning showed 89.02%, 82.52%, 85.64%, and 88.70% for average precision, recall, F1-Score, and accuracy, respectively, and Decision Tree had 85.42% average precision, 84.13% average recall, 84.77% average F1-Score, and 87.82% average accuracy. Random Forest, as the next best, showed 82.86%, 73.56%, 77.93%, and 83.15% for average precision, recall, F1-Score, and accuracy, respectively. KNN, again outperforming lowest-ranked MLP, showed 74.36% for average precision and 68.35% for average recall, with 71.24% and 78.38% for its F1-Score and accuracy, respectively. MLP, being the worst performer, had results of 65.33%, 51.83%, 57.80%, and 72.07% for average precision, recall, F1-Score, and accuracy, respectively.

Table 2.6: Average cross validation score of each model (20-min prediction validity with an 80% threshold).

Metrics	MLP	KNN	RF	DT	EL
Precision	65.33	74.36	82.86	85.42	89.02
Recall	51.83	68.36	73.56	84.13	82.52
F1-Score	57.80	71.24	77.93	84.77	85.64
Accuracy	72.07	78.38	83.15	87.82	88.70

MLP=Multilayer Perceptron, KNN=K-Nearest Neighbors,
RF= Random Forest, DT=Decision Tree, EL=Ensemble Learning

For a better view, Figures 2.4–2.7 present the graphical comparison of all the models. By analyzing all the experimental results, it is clear that, in terms of the evaluation metrics, Decision Tree and Ensemble Learning performed better than the other models. However, given the complexity of the Ensemble Learning approach (a combination of all the models), it did not show a significant improvement when compared to the Decision Tree model. When both computational complexity and performance are considered, Decision Tree was the optimized model throughout all of these experiments.

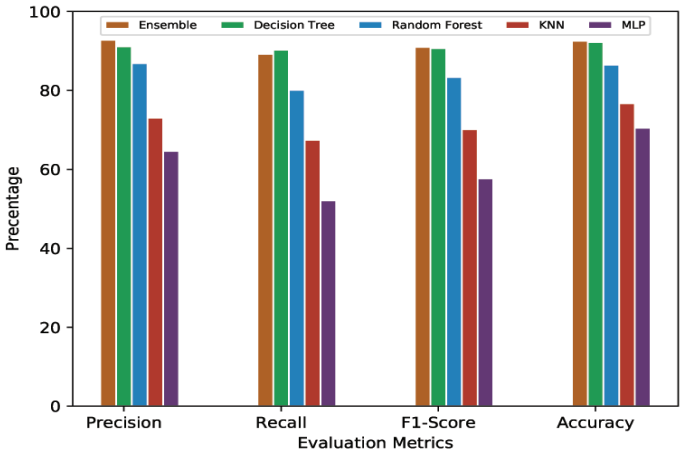


Figure 2.4: Graphical representation of comparative analysis of ML/DL approaches (prediction validity = 10 min, threshold = 60%).

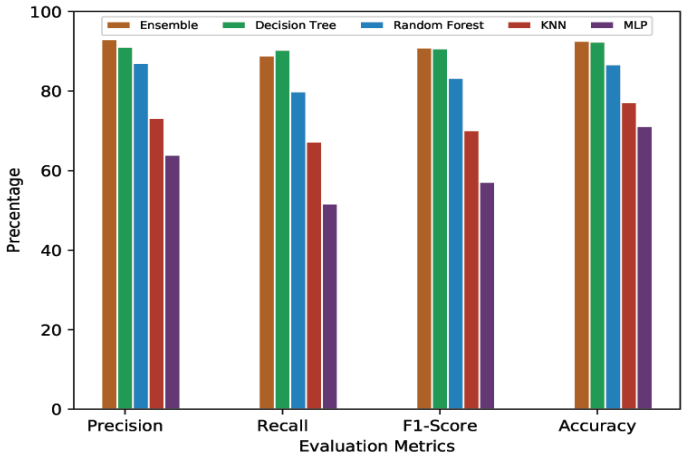


Figure 2.5: Graphical representation of comparative analysis of ML/DL approaches (prediction validity = 10 min, threshold = 80%).

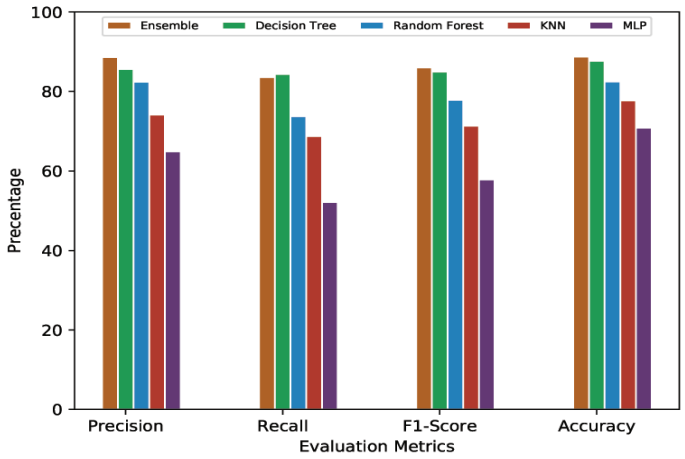


Figure 2.6: Graphical representation of comparative analysis of ML/DL approaches (prediction validity = 20 min, threshold = 60%).

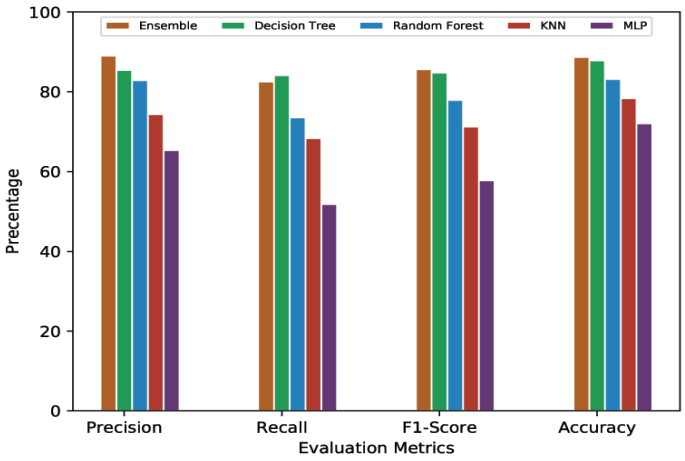


Figure 2.7: Graphical representation of comparative analysis of ML/DL approaches (prediction validity = 20 min, threshold = 80%).

2.5.4.5 Training Data Evaluation

The size of a training data set can significantly influence the performance of an ML/DL Model. Therefore, in order to further evaluate all five ML/DL models, we performed another comparison, designed to observe how the size of the parking space training data set affects the performance of these models. We chose a subset of the total data set containing 1,252,936 records. We partitioned this data set into five equal folds and set one of the folds as the testing data. Hence, each fold contains 250,587 records. To ensure better observations, we began training our models with a very small number of records: 1000 records. We then added the next 40,000 records and trained the models with 50,000 records. For the third iteration, we trained the models with the 250,587 records of one fold. Then, for the rest of the iterations, we added 250,587 records into the training data set to keep flow consistent. Not following the trend of the other iterations, for the first two iterations, training data set size was randomly chosen as very low (1000, 50,000) to observe how the models behave with very low training data size.

We evaluated the performance of each model in terms of accuracy, gradually increasing the training data size at each level (Figure 2.8). Figure 2.9 shows that, after 50,000 records, KNN and Random Forest have a constant, very low increase in accuracy, leading to very moderate improvement. In contrast, Decision Tree and Ensemble Learning showed a bit lower accuracy (around 64% and 68%, respectively) when 1000 records were used as training data. However, both of these models showed continuous improvement as more data were added to the training set. MLP, in contrast, showed a very low accuracy (around 28%) when 1000 records were used, and then only a negligible improvement (almost no improvement) from its accuracy at 50,000 records.

We conducted this experiment for the scenarios mentioned in the *Performance Evaluation* section and found a similar behavior throughout. This experiment, based on a subset of the data set, reveals the behavior exhibited by these models when used for the scenarios in the *Performance Evaluation* section.

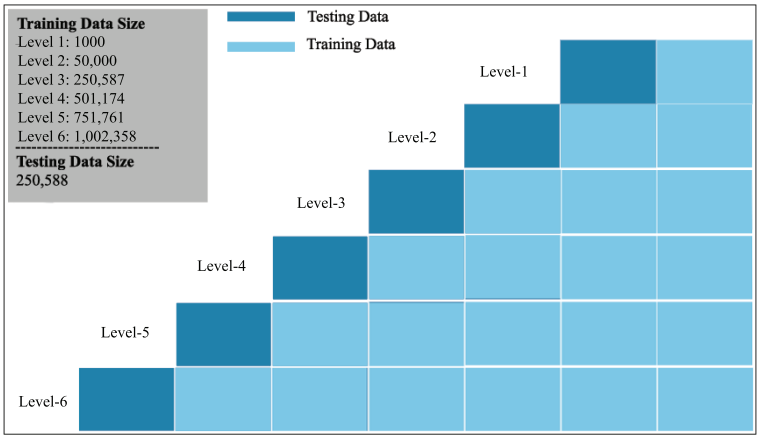


Figure 2.8: Training data size evaluation method.

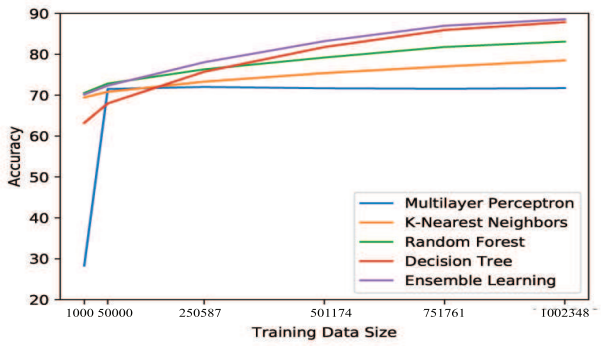


Figure 2.9: Performance evaluation of training datasize.

2.5.4.6 Distance Based Recommendation

Individual parking spot prediction enables us to recommend the parking spot to the user with respect to distance. As shown in Figure 2.10, all the parking spots, predicted as "available", can be sorted with respect to distance given position (coordinates) of vehicle and parking spots. Users cannot be given indications of parking spaces too far from each other. Thus, the calculation and the clustering of results should be organized by identifying some limited areas close to the final destination that have the highest probability to have free parking spaces. For the time being, no reservation capabilities nor differentiation between

prices of the parking slot have been considered; however, these are functions that could be easily added to a recommendation system.

In order to calculate the distance between vehicle and free parking spots and provide recommendation on the basis of distance, we use GPS coordinates of parking sensors & vehicle, and the well-known Haversine formula [36]:

$$a = \sin^2(\delta\theta/2) + \cos\theta_1 - \cos\theta_2 \sin^2(\delta\lambda/2), \tag{2.10}$$

$$c = 2a \tan 2(\sqrt{a}, \sqrt{1-a}. \tag{2.11}$$

$$Distance = R.c \tag{2.12}$$

In Equations (2.10)–(2.12), θ is latitude, λ is longitude, and R is earth's mean radius (i.e., 6371km). After calculating the distance between vehicle and free parking spots, sensors are sorted in ascending order (from nearest to farthest). A functionality, built on such calculation, can be used to recommend to users the closest parking slot with the maximum probability of finding it free.

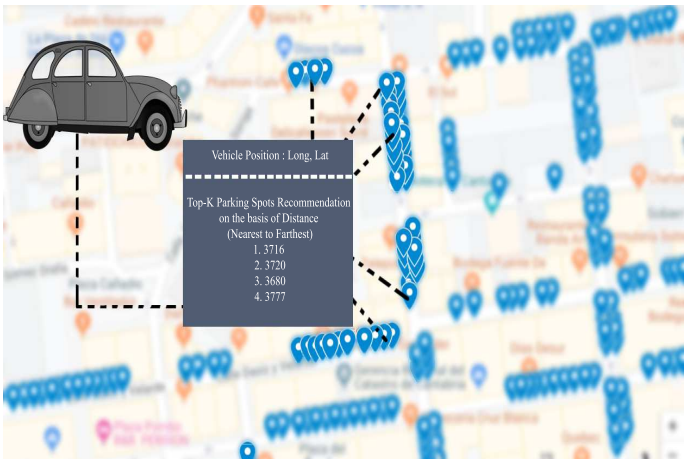


Figure 2.10: Recommending top-K parking spots on the basis of distance.

Chapter 3

Road Traffic Prediction Improvement using Air Pollution and Atmospheric Data

Contents

3.1 Overview

Traffic flow forecasting is one of the most important use cases related to smart cities. In addition to assisting traffic management authorities, traffic forecasting can help drivers to choose the best path to their destinations. Accurate traffic forecasting is a basic requirement for traffic management. We propose a traffic forecasting approach that utilizes air pollution and atmospheric parameters. Air pollution levels are often associated with traffic intensity, and much work is already available in which air pollution has been predicted using road traffic. However, to the best of our knowledge, an attempt to improve forecasting road traffic using air pollution and atmospheric parameters is not yet available in the literature. In our preliminary experiments, we found out the relation between traffic intensity, air pollution, and atmospheric parameters. Therefore, we believe that addition of air pollutants and atmospheric parameters can improve the traffic forecasting. Our method uses air pollution gases, including CO, NO, NO_2, NO_x, and O_3. We chose these gases because they are associated with road traffic. Some atmospheric parameters, including pressure, temperature, wind direction, and wind speed have also been considered, as these parameters can play an important role in the dispersion of the above-mentioned gases. Data related to traffic flow, air pollution, and the atmosphere were collected from the open data portal of Madrid, Spain. The long short-term memory (LSTM) recurrent neural network (RNN) was used in this chapter to perform traffic forecasting.

3.2 Introduction

Motivation

Vehicular traffic management is a major issue in cities and metropolitan areas [37]. Traffic has a relevant impact on different aspects of daily life, from time spent in the traffic jams to higher level of pollution produced, from gas and resources consumption to infrastructural investments and maintenance of road and transportation system [38]. Traffic management and optimization is an essential part in every smart city platform. Smart mobility is one of the most important services of smart city platform. It has a direct impact on the quality of life of citizens and on the ability of the city to support the exchange of people and goods within the urban environment. Traffic regulation and orchestration are key components. With a city's large number of vehicles, problems related to traffic are critical for the effective functioning of the city and the health of its citizens. Traffic congestion is a major problem, especially when it is associated with an increasing number of vehicles in use (e.g., in developing countries, or in cities with inadequate public transportation). It leads to environmental, social, and economic issues [39]. The timely prediction of traffic flow can be

helpful to avoid congestion, as drivers can choose the most comfortable and less congested path to reach their destination, or modify their time schedule for their journey in order to compensate for the expected time of arrival caused by the traffic. Road traffic forecasting is defined as the estimation or prediction of the traffic flow in the (near) future. Another aspect of traffic levels in cities is car and truck generated air pollution. Many cities suffer from air pollution. Increasing traffic emissions are one of the major contributors to urban air pollution [40]. According to the World Health Organization (WHO) [41], a large portion of air pollution is contributed by the transport sector. These two phenomena are linked, and many cities are tackling this problem by deploying sensors for measuring traffic intensity and air quality. Air pollution generated by traffic depends on several factors, ranging from the types of vehicles (gasoline, diesel, electric), to the level of congestion and the time spent in traffic jams, the atmospheric or geographical characteristics of the environment, and many more.

A large networks of sensors have already been deployed in several cities (e.g., Madrid, Santander, and Barcelona in Spain, Singapore, Seoul, Copenhagen). Data generated by these sensors are very useful for forecasting. For example, around 4000 traffic intensity sensors are deployed in Madrid, Spain (figure 3.1) [42]. These sensors provide information about the number of vehicles passing per hour (actually every 15 minutes). Similarly, there are 24 stations measuring air pollution (figure 3.2) and 26 stations collecting atmospheric data such as local temperature, pressure, wind speed, and wind direction (figure 3.3). Madrid's data, then, offer the possibility to further analyze the correlations between traffic intensity, levels of pollution, and meteorological condition. Figures 1 to 3 show that traffic intensity sensors are greater in number as compared to air pollution sensors. Air pollution sensor data are not so granular as the traffic intensity ones. Therefore, in our experiments, we chose traffic sensors in close proximity (upto 500m)(figure 3.10c) to air pollution sensors and, vice versa, we selected air pollution sensor stations close to big roads or crossroads. Air pollutants such a CO, NO, NO_2, NO_x, and O_3 are associated with road traffic [43] [44] [45]. The combination of large quantities of curated data with machine/deep learning models can provide useful insights for the correlation of traffic with air pollution. Many studies demonstrate how data about traffic flow can be used to predict air pollution. For example, Batterman et al. [46] used a dispersion model, called the Research Line Source (R-LINE) model, and emission inventory to predict the air pollutants $PM2.5$ and NO_x. Ly et al. [47] predicted the concentration of NO_2 and CO by using multisensor devices data and weather data, including temperature, relative humidity, and absolute humidity. In this work, they used the data of an Italian city (unnamed city) between March 2004 and February 2005. Similarly, Lana et al. [48] used a Random Forest regression model to predict the air pollution level with respect to road traffic utilizing open data from Madrid for the year

2015. Russo et al. [49] used atmospheric data, including temperature, wind direction, wind intensity, along with other air pollutants, including NO_2, NO, and CO as input variables to neural network to forecast the concentration of $PM10$. However, in their experiments, they did not take traffic intensity into account. Brunello et al. [50] investigated temporal information management to assess the relationships between air pollutants, including NO_2, NO_x, and $PM2.5$, and road traffic. In all of these studies, thanks to the direct link between road traffic and air pollutants, road traffic was used to predict air pollution. Air pollution and traffic intensity data are collected as time series of values and are generally made available for analysis and study. However, to the best of our knowledge, there has not yet been an attempt to use air pollution to improve the traffic forecasting. Traffic intensity is a major contributor to air pollution. The presence of certain pollutants in the air is most likely determined (or largely contributed) by vehicle traffic. Being able to correlate the actual level of these pollutants, on a timely basis for an area close to an air pollution station, to the expected level of traffic in the same area can be of help in better predicting the traffic intensity. Hypothetically, if the only source of pollution was car traffic, a strong correlation between the air pollution level and the intensity of traffic could be drawn. Cities and urban conglomerates are complex systems and there are other major contributors to air pollutions (home heating, factories and transformation implants, and others). Besides this, also meteorological condition can influence the air quality, e.g., strong winds can spread and disseminate pollutants in large areas making it more difficult to find strong correlations between traffic, air pollution and other contributors. In spite of the complexity of these causal relations, Madrid offers an impressive wealth of data for approaching and further study the correlation between traffic intensity and air pollution. The analysis considers the current level of pollution in a specific area at a specific time interval "t" as an evidence of presence of traffic. This evidence is also reinforced by the ability to know the traffic intensity levels before the time "t". Using these data could lead to a better prediction of the traffic intensity. Generally speaking, the approach of considering air pollution data as a means to predict traffic intensity can be undertaken in two ways: to use air pollution data together with traffic intensity data to improve the prediction of traffic intensity, or to use the air pollution data and numerical models to infer the expected traffic intensity. This chapter evaluates the first option, while the second one is left for further study.

Cities are systems that attract people, goods and activities and their impact is not limited to the city limits, but extend to cities, towns, and villages in the surrounding area. According to a World Economic Forum report [51], people prefer living, staying, studying, and growing up in cities. In fact, big cities exert a strong attraction effect and have a considerable impact on very large areas. The traffic and pollution issues involved may therefore be better analyzed if the extended areas are considered. Sometimes, air quality

measurements are also assessed in decentralized areas. Thanks to the availability of several open datasets, it is possible to investigate the correlation between air pollution and traffic intensity that may have contributed to the level of pollution in large monitored areas. This information will in turn offer the possibility to focus on air quality analysis and to correlate it to the expected traffic intensity. This chapter investigates this possibility, starting from a highly-sensed and populated area (Madrid and its surrounding area). In Madrid's data portal, datasets related to air pollution and atmospheric data are available timely each hour. On the other hand, data for traffic flow is updated each 15 minutes. Historic data of traffic flow, air pollution, and atmospheric variables for each month is made available at the end of the month. One expected outcome of this work is to validate (or reject) the usage of current air pollution measurements and levels combined with atmospheric data to improve the prediction of the traffic intensity levels.

Traffic intensity is the major cause of the pollution problem. So not surprising, measuring or using the resultant levels of pollution generated can be a means to understand how many vehicles may be present. Pant et al. [52] performed an analysis to characterize the traffic-related PM emissions in a tunnel environment. For this purpose, they chose 545 meters long, one of the major tunnels in Birmingham, called A38 Queensway Tunnel. Around 25000 vehicles travel through this tunnel daily. They deployed the PM sensors at the distance of 1.5 m on emergency layby. A similar experiment can be done with different number of vehicles to observe the volume of the pollution produced. A set of vehicles operating for a specific period of time in the same area will produce a very similar quantity of pollutants (imagine 100 cars in a closed environment, they will produce the same amount of pollutants when operating for the same period of time). Measuring the levels of pollutants over time may create a dataset usable to predict level of pollution as well as from the pollution levels to determine how many cars were contributing. Hypothetically, measuring the level of pollution at a certain instant may allow to determine how many cars were operating. In the real-world things are more dynamic, for instance:

- the concentration of pollutants is greater close to big roads [53] (this is also why we tried to consider traffic intensity sensors close to the pollution sensors).
- the set of vehicles may be dynamic in composition (more diesel, more electric, and so on) during the days.
- the pollution level generated can be impacted by the meteorological condition.

However, the traffic in a city shows patterns and in spite of the dynamic of the composition/aggregation of vehicles producing pollutants, there are patterns also in how people use the cars (e.g., similar number of commuters in peak hours of traffic). These patterns

are also well-known by users, they, in fact, expect to have different traffic condition during the day and the week (with large differences between working days and week-ends). Over a long period of time, these patterns repeat and the levels of pollution can be considered as signatures of traffic intensity. The hypothesis to verify is if the levels of pollution may correspond on the average to certain levels of traffic and if these measurements of pollution can be used to improve the traffic predictions. Having time series of the pollution signatures together with time series of traffic intensity will allow to better predict the traffic intensity.

The objective is also to determine if such an approach is practical and if it can give useful and improved results over an analysis that considers only the traffic intensity time series. Determining the relations between levels of pollution and traffic intensity may lead to important consequences such as: to better control the air quality in more parts of the city and still maintain the desired levels of monitoring of vehicular traffic situation; the reduction of the number of traffic sensors, which can lead to reduced maintenance costs that could go in favor of a more capillary environment management infrastructure; moving from specific sensing and monitoring to general-purpose sensing for large urban environments [54]; the integration and exploitation of other forms of environmental control (e.g., satellite data).

LSTM recurrent neural network is very popular for dealing with time-series data [55]. In the case at hand, the relationship between traffic intensity and pollution levels are aligned (Figures 3.5-3.6) which provides the indication towards possible correlation (whose extend needs further analysis), other time the relationship is blurred by other factors (e.g., meteorological factor). Neural Network can be fruitfully used to capture the evident and the more hidden patterns. For instance, in a week period different patterns (working days versus week-end may show different courses). An adequate period of time for a repeated number of time (e.g., a weekly observation for a duration of a year of data) may disclose relevant correlations. Therefore, we adopted a long short-term memory (LSTM) recurrent neural network (RNN)-based approach which uses air pollutants, including CO, NO, NO_2, NO_x, and O_3, along with some atmospheric variables including pressure, temperature, wind direction, and wind speed to improve road traffic forecasting in Madrid, Spain. The experiments presented in this chapter are based on one year of data collected from Madrid's open data source. Complete details about the dataset are provided in section 4.

Organization

The chapter is organized as follows. Section 3.3 offers a summary of the related work, and section 3.4 explains the methodology. The dataset information and performance evaluation are provided in section 3.5.

Figure 3.1: Traffic intensity sensors in Madrid

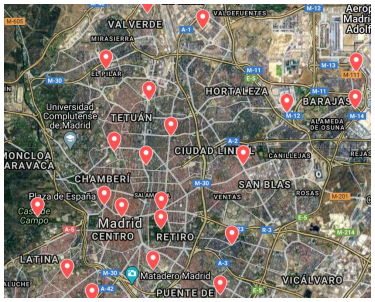


Figure 3.2: Air pollution sensors in Madrid

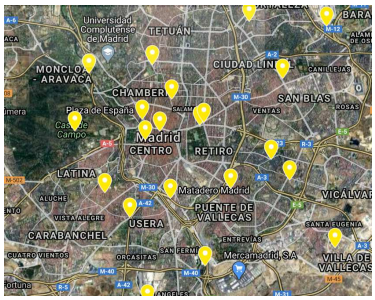


Figure 3.3: Weather stations in Madrid

3.3 Related Work

In this section, we summarize the existing work on traffic forecasting available in the literature. Ji et al. [56] used a deep learning, LSTM RNN-based model exploiting long-term evolution (LTE) access data as an input to their model for the prediction of real-time speed of the traffic. Similarly, Wei et al. [57] proposed an AutoEncoder and LSTM-based method to predict traffic flow. They collected data from the Caltrans Performance Measurement System (PeMS) and considered only three features: 1) traffic flow, 2) occupancy, and 3) speed. Li et al. [58], in their chapter, provide an overview of the machine learning approaches for short-term traffic forecasting. Ketabi et al. [59] provide a comparative analysis of multiple variant recurrent neural network and conventional methods for traffic density prediction. They used 40 day data, generated by 58 cameras in London, of the time slot between 9:30 AM and 6:30 PM. Their work considered two features: time and traffic density. Zhu et al. [60] used GPS information data to develop a traffic flow prediction model. Based on data clustering using historic GPS data, their artificial neural network-based prediction model utilized a weighted optimal path algorithm to predict short-term traffic flow. This prediction, based only on the departure time, was then used as input to an A-Dijkstra algorithm to find an optimal path.

Hou et al. [61] proposed a hybrid model that combines an autoregressive integrated moving average (ARIMA) algorithm and a wavelet neural network algorithm for short-term traffic prediction. Their experiment is based on a case study of the Wenhuadong/Tongyi intersection in Weihai City, and only considers weekdays. They collected data over three workdays, using the data from first two days for training and 3rd day's data for testing. Time and traffic flow were the only two features considered. Similarly, Tang et al. [62] proposed a hybrid model, comprising denoising schemes and support-vector machines for traffic flow prediction. To conduct their experiments, they collected data from three traffic flow loop detectors deployed on a highway in Minneapolis, MN (USA). They considered five

denoising methods (Empirical Mode Decomposition, Ensemble Empirical Mode Decomposition, Moving Average, Butterworth filter, and Wavelet) for performance evaluation purposes. Their data contained three features: volume, speed, and occupancy. Wang et al. [63] presented an integrated method, combining Group method of data handling (GMDH) and seasonal autoregressive integrated moving average (SARIMA), for traffic flow prediction in the Nanming district of Guiyang, Guizhou province, China. They collected data for five working days; data from the 1st four days were used for training while the last day's data were used for testing. They used residue series as features and labels, respectively to train the model. Rajabzadeh et al. [64] proposed an hybrid approach for short-term road traffic prediction. Based on stochastic differential equations, their approach ultimately improves the short-term prediction. They divided their approach into two steps: (1) a Hull-White model implementation to obtain a prediction model from previous days and (2) the implementation of an extended Vasicek model in order to model a difference between predictions and observations. Two datasets were used: one from a highway in Tehran, and the other an open dataset of PeMS time and traffic volume as inputs. Goudarzi et al. [65] proposed an approach based on self-organizing vehicular network to predict traffic flow. They used a probabilistic generative neural network technique, called deep belief neural networks, to predict traffic flow. Data generated by road side units (RSUs) were used for experiments, with traffic volume and time as inputs. Abadi et al. [66] used traffic flow series that indicate the trends in traffic flow; wavelet decomposition provided basis series and deviation series from the traffic flow data. In addition, local weighted partial least squares and Kalman filtering were used to predict the basis series. One day's data (8:00 AM to 8:00 PM) from the website of the ministry of communication of Taiwan were used for their experiments. Zhang et al. [67] used atmospheric data (average wind speed, temperature, ice fog, freezing fog, smoke) as input to gated recurrent neural network to predict the traffic flow. Rey del Castillo [68] presented an analysis on Madrid's traffic. In this work, short-term indicators of traffic evolution have been produced. Similarly, Lagunas [69] used different machine learning algorithms, including K-means, K-nearest neighbors, and Decision Tree, combined with traffic data, weather data, and data related to events in Madrid to predict the traffic congestion in an area.

The majority of the above-mentioned works used traffic intensity and time in order to forecast traffic. However, we believe that some other parameters like atmospheric conditions can effect the traffic flow which have not been considered in above-mentioned works. Tsirigotis et al. [70] considered only rainfall, along with traffic volume and speed to forecast the traffic. Similarly, Xu et al. [71] considered temperature and humidity, along with taxi trajectory data to forecast traffic flow. They took travel time, pick-up & drop time, and distance into account to forecast traffic flow. Only one month's data (01 Jan 2015 to

31 Jan 2015) were considered. We believe, traffic pattern can vary in different days and months. For example, we might observe different traffic pattern during weekends. Similarly, according to a case study in Copenhagen, Denmark, 80% journeys are made on foot in city center and 14% are made by bicycle in summer [72]. On the other hand, traffic forecasting based-on taxi trajectory might have other flaws too. For example, road lines leading to airports might have heavy traffic flow as compared to other lines in surrounding areas. Traffic forecasting for surrounding areas, based on taxi traveling in the lines with heavy traffic flow might result an inaccurate forecasting. In this chapter, we are introducing the use of air pollutants and atmospheric parameters (pressure, temperature, wind direction, and wind speed) to forecast traffic. These are the two motivations for using atmospheric parameters: they influence the level of air pollutants in the air, and they also can influence the human behavior. For example, Badii et al. [19] used weather conditions, including temperature, humidity, and rainfall to predict the availability of parking spots inside parking garages, given the fact that depending on the weather condition, people's choice of parking may vary. For example, in thunderstorm, people will prefer indoor parking. Similarly, on different occasions, people may prefer to use public transport which may affect the occupancy of parking lots.

3.4 Methodology

In this section, we describe the methodology for forecasting traffic flow using traffic intensity values. A first step was to use traffic intensity data combined with air pollution and atmospheric data in order to forecast the traffic. We correlate traffic intensity data to air pollution and atmospheric variables (as we also want to study the relationship between traffic and pollution). As described earlier, air pollutants are often linked to the road traffic levels. Using that link, we propose to use air pollutants and atmospheric variables to forecast the traffic flow. In the second step, we used only time-stamped traffic intensity data, excluding air pollutants and atmospheric data, to forecast the traffic flow. The results produced from step one and step two were then compared to observe how air pollution and atmospheric data, combined with traffic intensity data, could be used to forecast traffic flow. Our experiments were organized into two categories:(1) statistical analysis and (2) traffic forecasting using LSTM RNN. For our experiments, we used open data, collected by the city of Madrid, Spain [73]. The first category of experiments was instrumental for analyzing the quality of available data and to identify macroscopic properties of the data sets.

3.4.1 Statistical Analysis

As the initial step, we chose one of the air pollution measuring stations and selected two traffic flow sensors at different distances (Figure 3.4).

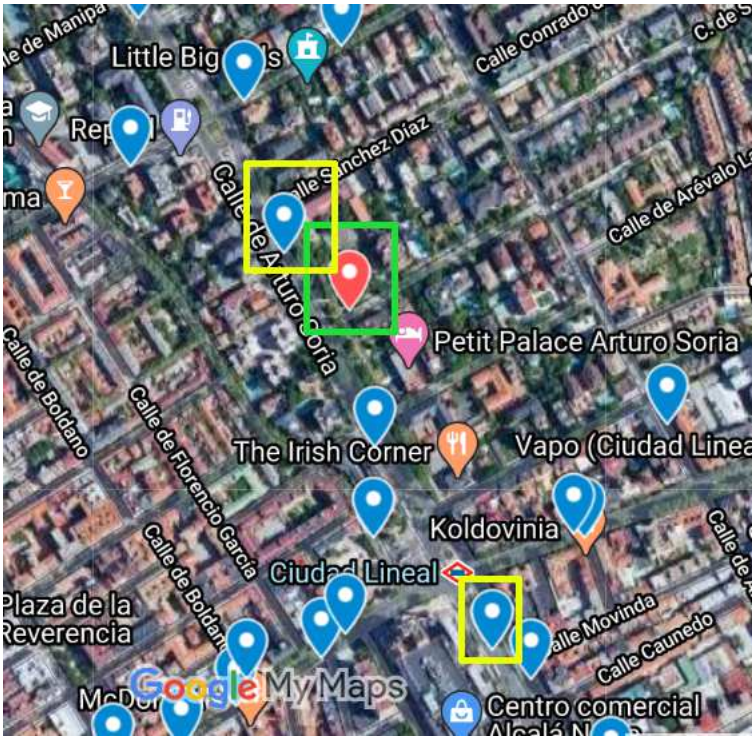
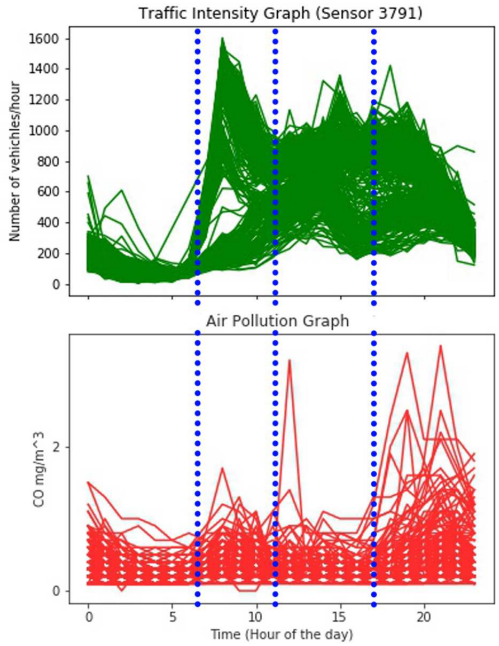
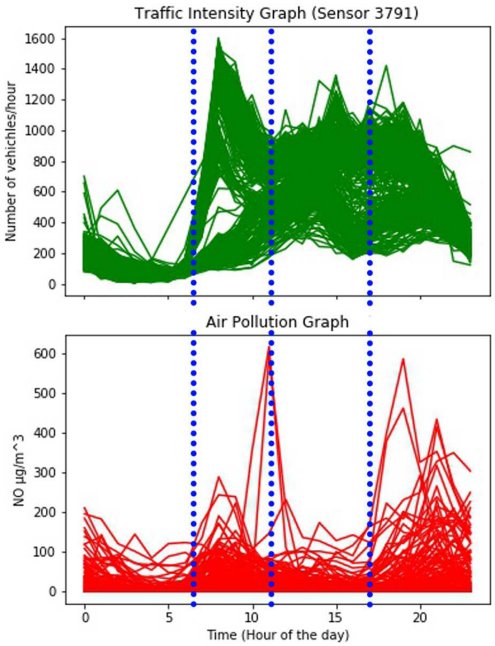


Figure 3.4: Considered air pollution station (highlighted by the green rectangle) and traffic flow sensors (highlighted by the yellow rectangles).

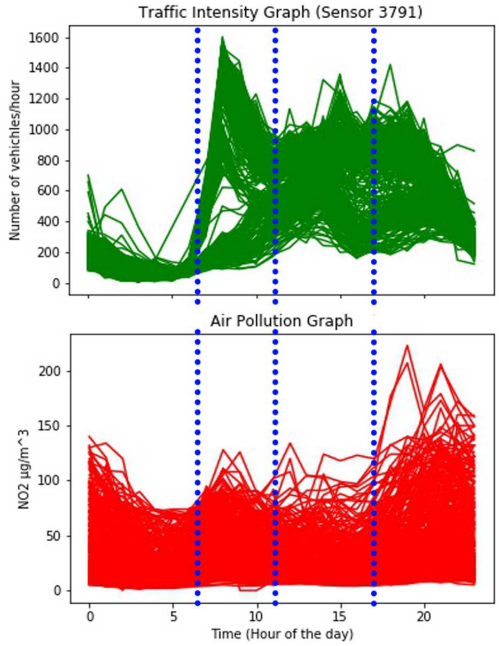
We collected hourly data from 01 January 2019 to 31 December 2019. This data contained the number of vehicles per hour that passed the sensors, and the air pollutants ($CO, NO, NO2, NOx$, and $O3$) levels. Subsequently, we used the accumulated data in order to have an initial view on the possible correlations and to determine a set of parameters that could have an impact on the correlation. We plotted the data on graphs in order to observe the traffic flow patterns with respect to air pollution, as shown in Figure 3.5. Figures 3.5 (a), 3.5 (b), 3.5 (c), and 3.5 (d) represent the hourly graph of traffic flow measures of one of the selected traffic flow sensors with respect to air pollutants CO, NO, NO_2, and NO_x.



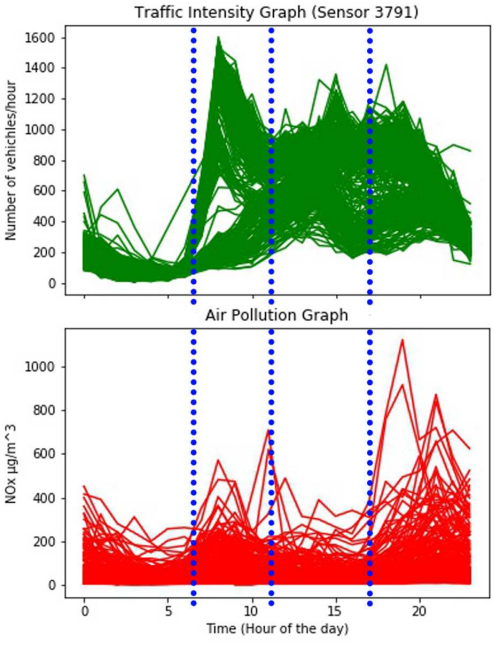
(a) CO to traffic flow correlation graph



(b) NO to traffic flow correlation graph



(c) NO2 to traffic flow correlation graph



(d) NOx to traffic flow correlation graph

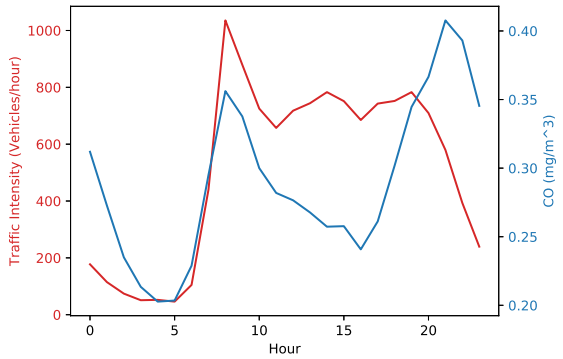
Figure 3.5: Correlation graphs of traffic flow and air pollutants with respect to each hour of the day.

These graphs represent the values of each hour of each day of the year 2019. The graphs

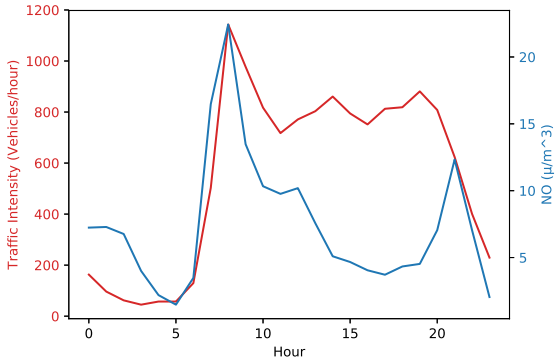
in green represent the traffic intensity while the corresponding graphs in red represent the air pollutant levels. In these graphs, blue dotted lines divide the graphs into four time intervals. During the first 2 intervals, all the measured air pollutants follow the traffic flow trend, with few exceptions. In the first interval, the pollutant levels decrease when the traffic is decreasing. Similarly, during the second interval, the pollutant levels increase when the traffic is increasing. A similar pattern can be seen during the fourth interval. However, during the 3rd interval, the pollutants do not seem to be following the traffic flow pattern. To investigate this phenomenon, we studied air pollution dispersion aspects and considered wind speed as one of the factors in air pollution dispersion [74]. Hence, as a further verification, we plotted a graph representing the average annual wind speed for each hour (Figure 3.7), which reveals that wind speed is constantly increasing during the time interval when air pollution does not follow the traffic flow pattern. Given the air pollution dispersion values and the available data, we consider that wind speed is one of the factors that influence air pollution dispersion. As mentioned above, we noticed from traffic-pollution visual pattern analysis that there are similarities in the growth of traffic and the growth of pollution during the morning, and there is a shift in the growth of traffic and the growth of pollution during the evening. In the mid of the day, the correlation is more difficult to capture. This is why we used RNN in order to determine some correlations beyond the statistical ones. The same algorithm using only traffic intensity data and using traffic intensity + meteorological + pollution data show different levels of precision in favor of the analysis that considers more contextual information (a comparative analysis is provided in the section 4.2). Figure 3.5 presents the correlation between air pollutants and traffic intensity with respect to each hour of each day of the year. However, in order to provide more insights related to correlation, we have plotted an annual mean graphs for all the considered air pollutants (Figure 3.6). Phase shift can be seen in Figure 3.6 too, however, phase shift in Figure 3.6 is different than that of in 3.5 because of average annual values.

3.4.2 Linear Interpolation

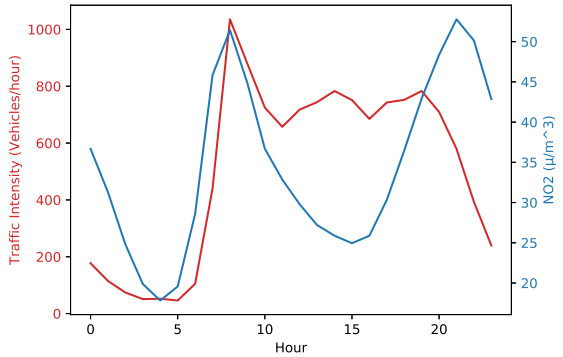
Missing values from the data is another major issue when dealing with time-series data. Even though the available open data of the city of Madrid is well maintained, minor glitches in sensors are almost inevitable.



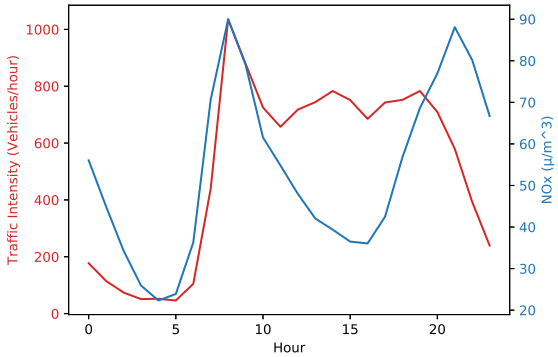
(a) CO to traffic flow correlation graph (annual mean)



(b) NO to traffic flow correlation graph (annual mean)



(c) NO2 to traffic flow correlation graph (annual mean)



(d) NOx to traffic flow correlation graph (annual mean)

Figure 3.6: Correlation graphs of traffic flow and air pollutants with respect to each hour of the day (annual mean).

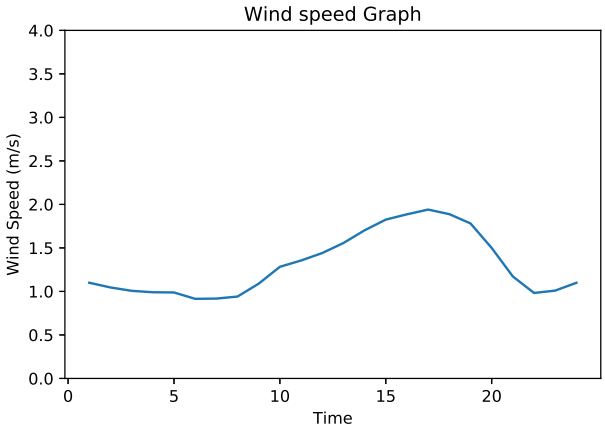


Figure 3.7: Average annual wind speed.

Sensors may go offline because of technical issues, or there is a possibility that received data could not be stored on a server. While conducting our initial data analysis, we observed that some of the traffic flow sensors had missing values for some timestamps. Though these missed values were not numerous, it was necessary to fill the gap because we were dealing with time-series data. In order to deal with this issue, we used a well-known method, linear interpolation. Linear interpolation is a popular technique to fill the missing values in a dataset [75]. This technique seeks to identify timestamps that are similar to those that are missing their values, and fills each missing value with an average value [76]. Linear interpolation states that there is a constant gradient in the rate of change between one sample point and the next point. Considering this assumption, if the amplitude of the i^{th} point is x_i and the amplitude of the $i+1^{th}$ point is x_{i+1}, then keeping the constant gradient, the j^th point between x_i and x_{i+1} can be calculated as follows [77]:

$$\frac{x_{i+1} - x_i}{(i+1) - i} = \frac{x_j - x_i}{j - i} \tag{3.1}$$

or

$$x_j = (j - i)(x_j - x_i) + x_i \tag{3.2}$$

3.4.3 Traffic Forecasting Using LSTM Recurrent Neural Network

When dealing with time-series data or spatial temporal reasoning, the LSTM RNN is considered one of the best options. As shown in Figure 3.8, unlike traditional neural networks, the LSTM RNN has memory units instead of neurons. With traditional fully connected neural networks, there is a full connection between the neurons of two adjacent layers. However, there is no connection between the neurons within the same layer. This lack of connection in traditional neural networks could create problems, and may likely cause total failure in terms of spatial temporal reasoning [78]. In RNNs, a hidden unit (memory unit) receives the feedback. This feedback goes from previous state to the current state. We used $timestamp$, $day\_of\_the\_week$, CO, NO, NO_2, NO_x, O_3, $pressure$, $temperature$, $wind\_direction$, $wind\_speed$, and $traffic\_flow$ as the features for our RNN. If we denote the input for the model as $x = (x_1, x_2, x_3, ..., x_T)$ and the output as $y = (y_1, y_2, y_3, ..., y_T)$, with the T in x and y is the prediction time, the traffic flow prediction at time t can be calculated iteratively using the following equations [79]:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \tag{3.3}$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) \tag{3.4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \tag{3.5}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \tag{3.6}$$

$$m_t = o_t \odot h(c_t) \tag{3.7}$$

$$y_t = W_{ym}m_t + b_y \tag{3.8}$$

In the above equations, $\sigma()$ represents the sigmoid function, which is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

and the $\odot$ in equations 3.3 to 3.8 represents the dot product (also known as scalar product). A memory block, shown in Figure 3.9, has an input gate, an output gate, and a forget gate. The output of the input gate is represented as i_t, that of the output gate as o_t, and the output of the forget gate as f_t, where c_t and m_t represent the cell and memory activation vectors, respectively. Similarly, W and b represent the weight and the bias matrix which are used to establish connections between input layer, memory block, and output layer. $g(x)$ and $h(x)$ are centered logistic sigmoid functions.

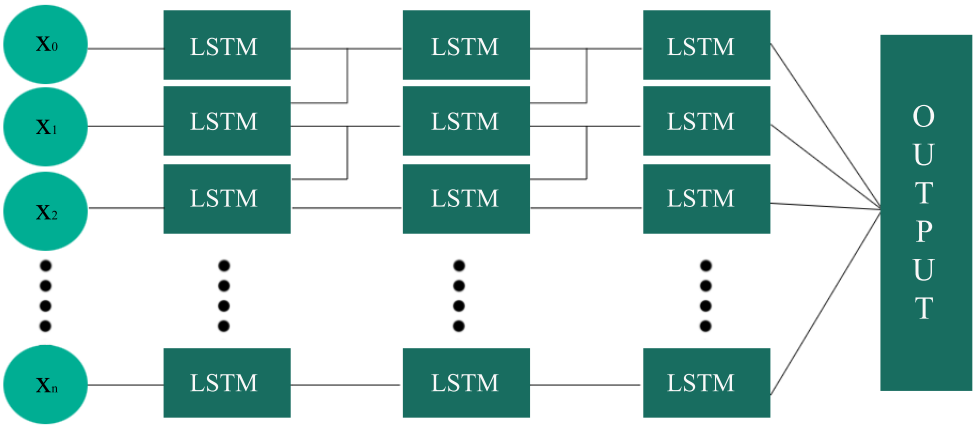


Figure 3.8: LSTM Recurrent Neural Network Architecture.

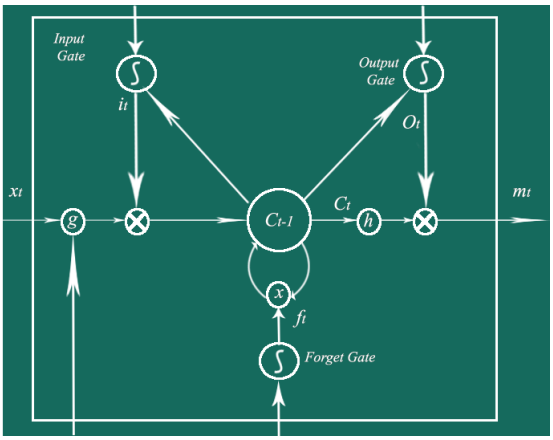


Figure 3.9: Architecture of a LSTM Memory Unit in Hidden Layers.

3.4.4 Data Normalization

Data normalization is one of the most important steps in data pre-processing. It guarantees the quality of the data before we use as the input to machine/deep learning models [80]. Data normalization is required when features have different ranges of values. For example, in our dataset, the traffic intensity values range approximately between 0 and 1500 while the value ranges for CO and NO_2 are 0–3.4 and 0–616, respectively. This difference of scale may lead to the poor performance of a machine/deep learning model. Data normalization helps to deal with data that contains values that have different scales. Moreover, it also helps to reduce the training time. Different kind of data normalization techniques are available, including min-max, median normalization, and Z-score decimal scaling. In this chapter, we used the most popular normalization technique, min-max normalization [81].

3.4.4.1 Min-Max Normalization

Min-max normalization maps data into pre-defined ranges i.e., [0,1] or [-1,1]. The values of each attribute in the data are defined according to their minimum and maximum value. If we denote the attribute in the data by "Atr", its value by "a_val", its normalized value as "a_norm", and pre-defined range as $[lower\_lim, higher\_lim]$, then following equation [80] can be used to calculate normalized values between the range $[lower\_lim, higher\_lim]$:

$$a\_norm = lower\_lim + \frac{(higher\_lim - lower\_lim) \times (a\_valu - min(Atr))}{max(Atr) - min(Atr)} \tag{3.10}$$

3.4.5 Hyperparameter

We used the following configuration of a LSTM RNN to forecast traffic flow using Madrid's open data:

- 3 LSTM layers;
- **Dropout:** To keep our model from going into overfitting, we applied dropout [82] at each LSTM layer with a value of 0.7;
- **Early Stopping:** To stop the training before the model approaches overfitting, we used early stopping [83] with the patience value of 5;
- **Look Back Steps:** In order to do prediction at time t, "look back" shows how many previous time steps need to be considered. We set the "look back" steps value at 168, which represents the total number of hours in a week. We chose 168 hours (one week) as "look back" period. The plan is to capture the evolution of the air pollutants over a period in which different, but recursive patterns may occur, e.g., working day traffic vs. Week-end traffic. We wanted to grasp the differences between working days and week-end. In addition, in such a period, the pollutants have time to consolidate (some pollutant can float for hours or more). Moreover, this time period could result a better forecasting. Traffic intensity shows different patterns between weekdays and weekends. Pollution "signatures" refer to longer and more complex situations. A week within a particular month (e.g., December before Christmas time) can be characterized by higher volume of traffic and hence pollution. Different months can have very different levels of traffic and pollution. The choice of considering one week is due to the possibility to grasp these variations, while still maintaining a short period for observation and data capture. With respect to pollution, a longer period of time (e.g., a month) would allow a more specific characterization of the traffic in that specific month and the related pollution signature could be used in order to help the prediction. A shorter period of time (one day, two days) is not able to capture these variations in traffic intensity and pollution measurements. However, the choice of one week is a starting point and, for further work, a better tuning of the time could be envisaged.

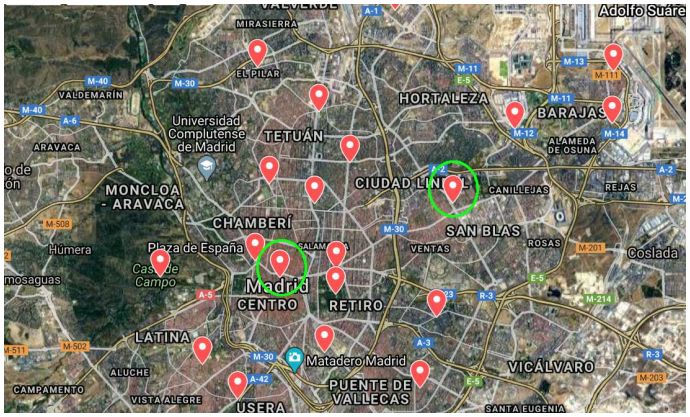
3.5 Dataset and Performance Evaluation

This section describes the dataset and its features, and evaluates the performance achieved by LSTM RNN for traffic flow forecasting using air pollution and atmospheric data. Open data from Madrid, Spain [73] collected and normalized for 1 year of observations. A large set

of data related to traffic intensity was collected in the first step. This dataset also contained weather and pollution-related features. We conducted experiments using the data from two air pollution sensor stations (Figure 3.10a) to forecast traffic flow. These stations measure CO, NO, NO_2, NO_x and O_3 values in the air. In addition, we used timestamp, traffic intensity, and atmospheric data, including temperature, pressure, wind speed, and wind direction from nearby weather stations. For a comparison, in the second step, we only used traffic intensity and timestamp values (with no air pollutant or atmospheric parameters) to forecast the traffic flow, and compared the results to see the effect of considering air pollutant and atmospheric data.

(a) Two air pollution sensor stations, considered for experiments.

(b) Traffic intensity sensors used for one air pollution sensor 28079016.



(c) Ariel view of Madrid's map showing the areas considered within 500m radius of both air pollution sensor stations.

Figure 3.10: Considered air pollution sensor stations, traffic intensity sensors, and areas in Madrid.

We chose 25 traffic flow sensors in a 500_m radius of the two air pollution sensor stations (Figure 3.10b). Traffic flow data is available after every 15 minutes, however, other data,

including CO, NO, NO_2, NO_x, O_3, *Pressure*, *Temperature*, *Wind Speed*, and *Wind Direction* are updated hourly. As the air pollutant data and atmospheric data are available hourly, therefore, we collected the hourly traffic data to keep it coherent with air pollution and atmospheric data. Table 3.1 represents the details of the features used to train the model. As our data were organized hourly (from 01 January 2019 to 31 December 2019), we had 8760 records in total; 67% of our data were used for training and 33% were used for testing. In order to extract the traffic flow insights for the roads where sensors are deployed, Table 3.2 represents the statistics of 25 traffic flow sensors within the chosen distance from the associated air pollution sensor station, and the minimum, maximum, and average traffic flow in the year 2019. Out of 25 sensors, 9 were faulty and gave either null value or garbage values. For those sensors, the minimum, maximum, and average flow values are represented as "NA" in Table 3.2.

3.5.1 Evaluation Metrics

In order to evaluate the results of the experiments, we defined some metrics to be used for the evaluation of our model.

Table 3.1: Features used for training the model.

Feature	**Value/Unit**
Month	1-12
Day	1-28/29/30/31
Weekday	1-7
Hour	0-23
CO	mg/m^3
NO	$\mu g/m^3$
NO_2	$\mu g/m^3$
NO_x	$\mu g/m^3$
O_3	$\mu g/m^3$
Pressure	mb
Temperature	$C°$
Wind Direction	Angle
Wind Speed	m/s
Traffic Flow	Vehicles/hour

Table 3.2: Traffic flow sensors' statistics.

Air Pollution Sensor Station	Traffic Flow Sensor	Distance from Air Pollution Sensor Station	Minimum Flow (Annual)	Maximum Flow (Annual)	Average (Annual)
28079016	6037	240m	0	384	112.344
	3791	79m	4	1601	493.693
	3775	294m	17	1166	468.615
	5938	205m	0	220	32.011
	5939	125m	5	1980	522.943
	10124	242m	NA	NA	NA
	6058	214m	NA	NA	NA
	3594	296m	NA	NA	NA
	5922	366m	NA	NA	NA
	10128	500m	4	1413	437.701
	10125	455m	NA	NA	NA
	5941	303m	0	1324	135.017
	5923	426m	5	1334	437.864
	5994	483m	0	480	135.389
	5940	369m	NA	NA	NA
	5942	336	0	1523	534.091
	5944	349m	0	182	72.176
	5921	374m	23	1214	481.669
	3776	425m	17	1208	476.911
	5937	484m	0	313	86.216
28079035	3731	26m	NA	NA	NA
	4303	39m	0	181	52.188
	3730	133m	NA	NA	NA
	4301	137m	NA	NA	NA
	10387	196m	40	1260	608.482

We used two of the most-used evaluation metrics *Mean Absolute Error (MAE)* and *Means Squared Error (MSE)*. Their mathematical representations are [84, 85]:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i^{predicted} - y_i^{observed}| \tag{3.11}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y^{predicted} - y^{observed})^2 \tag{3.12}$$

MAE is not sensitive to outliers. It does not deal well with big errors. It is very useful for continuous variable data. MSE is very useful when the dataset contains outliers. At

the beginning of the analysis, we wanted to be sure to grasp insights from very different data and patterns (traffic intensity and air pollutants). For this reason, we decided to check our results using both MSE and MAE. However, in our case, we found out that MAE alone could be used to evaluate the whole performance. Therefore, in future work, for additional experiments, we will use MAE for the evaluation. We used the training loss and the validation loss in the learning curve in order to be sure that our model was not overfitting.

3.5.2 Results

This section provides the MAE and MSE scores of the LSTM RNN model for each of the operational traffic flow sensors (excluding faulty sensors). As explained in the previous section, 25 traffic intensity sensors were considered, and out of those 25, 9 sensors were faulty and so were eliminated from the dataset during the experiments. Hence, Table 3.3 presents the MAE and MSE scores of 16 traffic flow sensors. We performed an hourly forecast. In order to do that, we determined the traffic intensity at time t by considering traffic intensity data, air pollution data, and atmospheric data from $[0, t-1]$ and, air pollution data and atmospheric data from time t.

The maximum MAE produced by the LSTM RNN for the traffic sensors within the radius of $500m$ of air pollution sensor "28079016" was 0.214 while the minimum MAE was 0.061. Similarly, the maximum MSE was 0.60 and the minimum MSE was 0.009. In order to evaluate our LSTM RNN model further, we conducted the same experiments for air pollution sensor station "28079035" and 5 traffic flow sensors within its $500m$ radius. Out of those 5 traffic flow sensors, 3 were faulty. Hence, Table 3.3 presents the values of 2 of the operational traffic flow sensors (4303 and 10387) around the station "28079035". The LSTM RNN produced values 0.105 MAE and 0.017 MSE for traffic flow sensor "4303", and 0.136 MAE and 0.029 MSE for traffic flow sensor "10387".

In order to observe the effect of introducing air pollutants and atmospheric parameters, we randomly selected five traffic intensity sensors and performed forecasting, considering only timestamped traffic intensity values. Figures 3.11 and 3.12 represent the comparative analysis of the mean absolute error and the mean squared error, respectively, with and without using air pollutants and atmospheric parameters as input features. It is clear that air pollutants and atmospheric parameters improve the MAE and the MSE. Our LSTM recurrent neural network-based approach performed better for all of the five considered traffic intensity sensors when air pollutants and atmospheric parameters were used along with the timestamped traffic intensity values.

Table 3.3: Mean absolute error (MAE) and mean squared error (MSE) for two considered traffic flow forecasting for considered traffic flow sensors.

Air Pollution Sensor Station	Traffic Flow Sensor	MAE	MSE
28079016	**6037**	0.183	0.045
	3791	0.206	0.056
	3775	0.206	0.054
	5938	0.073	0.009
	5939	0.166	0.035
	10128	0.203	0.053
	5941	0.061	0.005
	5923	0.188	0.046
	5994	0.173	0.047
	5942	0.214	0.060
	5944	0.208	0.056
	5921	0.200	0.051
	3776	0.193	0.051
	5937	0.160	0.030
28079035	**4303**	0.105	0.017
	10387	0.136	0.029

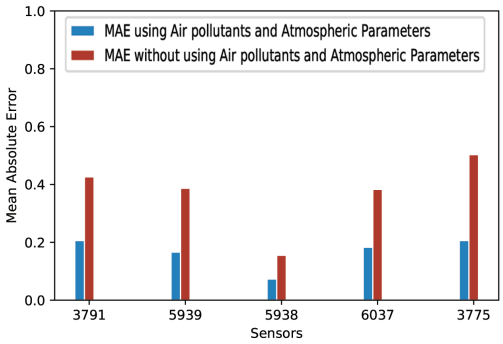


Figure 3.11: MAE with and without using air pollutants and atmospheric parameters.

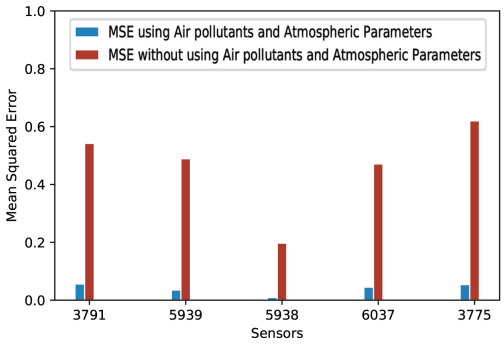


Figure 3.12: MSE with and without using air pollutants and atmospheric parameters.

3.5.3 Further Evaluation

To further evaluate the LSTM RNN model, we determined if our model was overfitting or not. One of the most-widely used methods for verifying overfitting [86] [87] is to plot learning curves. A learning curve plots a model's training loss and validation loss. These

curves give information about overfitting and underfitting:

- **Overfitting** represents the ability of the model to learn too much during the training process, so that when unseen data are provided for prediction, it shows poor performance. Overfitting can be diagnosed by plotting learning curves. If the training loss is decreasing but validation loss starts increasing after a specific point, this shows that a model is overfitting [87].
- **Underfitting** represents the inability of the model to learn from training data. If a learning curve shows either of the following two behaviors, the model is underfitting:
  - Validation loss is very high and training loss is flat regardless of training time.
  - Training loss is continuously decreasing without being stable until the training is complete.

Given above definitions, we plotted learning curves to observe the behavior of our model. Figure 3.13 shows that the learning curve of our model is not following any of the above-mentioned definitions of overfitting and underfitting. Training loss is decreasing and after a specific point it becomes stable. Similarly, validation loss becomes stable and remains close to the training loss. Both of these observations show that our model is a good fit.

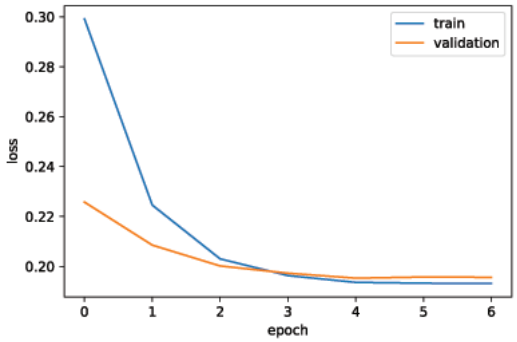


Figure 3.13: Learning curve representing training and validation losses of the LSTM RNN model for traffic flow forecasting.

3.5.4 Threat to Validity

The model utilized with the currently available data in Madrid. The penetration of electric vehicles may be a factor impacting the generation of pollution in major cities. This could have also a long term impact on our forecasts. However, the substitution of older vehicles with hybrid or electric ones will be relatively quick but not immediate. This delay will give the model some time to adapt and learn the new patterns. Given the ongoing concerns

about air pollution, the use of electric vehicles is increasing around the world. For example, the national electric mobility mission plan is anticipating the sale of around 7 million electric vehicles yearly from 2020 onwards [88]. While it will take a long time to completely eliminate conventional vehicles, the elimination of conventional fuel vehicles could be a threat to our approach's validity, as it is partially dependent upon vehicular pollution emission.

Chapter 4

Using Noise Pollution to Improve Traffic Prediction

Contents

4.1 Overview

Traffic prediction is one of the most important use cases for smart cities. Accurate traffic information is key to managing traffic issues. Many approaches that use traffic time series data to predict traffic flow have been proposed. In addition to traffic- specific parameters, some other features (called signatures) may be associated with road traffic, i.e., air and noise pollution. In this chapter, we show how noise pollution and traffic time-series data were used to train Long-Short Term Memory (LSTM) Recurrent Neural Networks (RNNs), which led to better traffic prediction on major roads in Madrid. This approach has already been used with pollution signatures. This work addresses a new potential investigation path closely related to the use of signature profiles and Artificial Intelligent techniques as a way to reduce the specialization of sensing infrastructure.

4.2 Introduction

Internet of Things (IoT) is a fundamental enabler for measuring and collecting data that represent physical phenomena in specific environments. This is the case, for example, of smart cities. Several initiatives [89] [90] are ongoing, designed to make the City and its environment first measurable, and then controllable. Sensing is usually oriented to monitor specific phenomena, e.g., traffic intensity, pollution and air quality, public transportation systems and the like. Sensing infrastructure is therefore deployed to measure specific features. Several sensing capabilities are needed (e.g., traffic intensity, air quality, and others). While they may use the same communications network, capabilities are not necessarily interoperable or related, and each one operates in an independent manner. The deployment of a smart city sensing infrastructure requires careful planning and a significant investment in infrastructure as well as in maintenance and operation.

A large body of literature is available on the best practices and analysis for optimizing the deployment and usage of sensing capabilities [91] [92] [93]. Typically, smart cities deploy "dedicated" infrastructure [94] comprising both communications and sensing capabilities. Sometimes the communication is "general-purpose", i.e., can be used by different specialized sensors. The management and operation of these infrastructures is another major cost. In fact, different sensor types and deployments (e.g., pollution vs. traffic intensity sensors) require different maintenance and operational capabilities. This diversity offers the possibility to collect precise data, but at the cost of differentiated management and operation processes. Non-dedicated sensing networks [94], i.e., sensing capabilities made available by users and other organizations, can be considered and integrated in a city's infrastructure. However, they may lack precision, stability and reliability and they almost always introduce integration issues. In this chapter, a novel approach is assumed: the availability of a

"general-purpose" sensing infrastructure capable of collecting a set of basic measurements from which it is possible to derive, by means of data fusion and artificial intelligence techniques, meaningful information about several phenomena occurring in a city. This approach has been proposed in [54] and [95] for smart city and enterprise environments, respectively. The context of smart cities is a challenging one, and this approach is still to be proven and validated. For instance, the identification of a basic set of sensing functions valid in an urban environment from which to infer as much reliable information as possible is evidently remains an open issue. However, if this approach is proved viable, creating a "general-purpose" network suitable for smart cities becomes a real possibility.

A homogeneous large sensing infrastructure could be created and maintenance and operational activities could be optimized, resulting in operation and cost savings. In addition, the possibility of moving complexity from the hardware infrastructure to the software layer will likely reduce costs and could increase the reusability of data for several different purposes in the city life-cycle, thereby breaking the silos of different data-sets and related sensor networks. The concept of the signature of a phenomenon can be introduced as the measured combination of basic sensed data strongly associated to an event, e.g., the pollution signature of traffic. This chapter focuses on an initial step towards this approach. As such, it shows how to use signatures (e.g., noise information) to better predict traffic intensity. The general idea is to determine pollution and noise signatures, i.e., characteristic profiles and levels of phenomena, that are strongly related to traffic levels. This approach is especially useful as a means to improve the current traffic predictions (correlating different data sets) and to verify if a step towards synthetic sensing for smart cities is possible.

If it is proved viable, this approach offers the possibility of using a general communications infrastructure for connecting a large number of well-structured and widely distributed general purpose sensing capabilities. The urban space can then be monitored and measured in a uniform way and a large amount of information can be extracted by means of AI techniques. In addition, the granularity of the "sensing" can be modulated to optimize the distribution and the deployment of sensors. This research has adopted a pragmatic approach in pursuing the objectives of identifying different data sets and investigating their relations. These data sets are measurements made available from large urban environments and contain actual data measured in the field. In fact, some cities collect and make available well-formed and complete data in the public domain [73] [96] [97]. To investigate complex relationships, some of these measures must be correlated, e.g., traffic intensity can have an influence on air quality. Data correlation [98] and interpolation [99] are emerging as techniques with which to infer good quality data in spatial and temporal environments/situations not fully covered by sensor networks alone. These datasets have been used to determine the correlations between different phenomena in order to improve the traffic

intensity predictions on real data.

Impact Statement

Smart Cities deploy large infrastructures for monitoring a variety of phenomena occurring in the urban space. They often target specialized and segmented tasks (e.g., monitoring the traffic). They are expensive; each of them requires specific management and they are not necessarily well accepted by citizens because they scrutinize an investment on specific phenomena that are not (usually) of general interest. Synthetic sensing promotes the movement of complexity from hardware to software infrastructures, unleashing new opportunities in terms of services to citizens. This approach, if proved viable, can reduce deployment and management costs, increase the functions and services offered, and help reduce the public's hesitance towards Smart Cities/the Smart Cities concept. For example, a wide deployment of pollution sensing devices would be politically, socially, and ecologically more acceptable than deploying specialized traffic sensors, especially if, by means of synthetic sensing and data fusion, correlated actionable information about other phenomena (e.g., traffic) could be derived at a fraction of the cost. From a technical perspective, synthetic sensing and data fusion applied to smart cities are rich with opportunities and possible new research paths. To begin, the identification of a basic set of non-invasive sensing features, followed by the definition of the limits of the information extraction from basic and raw data. Further on, the research could focus on identifying the best patterns for deployment: dense and granular for capturing more data, or sparse in order to save costs and reduce initial investments. Other promising fields are related to situation awareness and the introduction of Digital Twins as discussed in [100].

The rest of the chapter is organized as follows: section 4.3 provides the background, section 4.4 presents the steps carried out to train LSTM RNN to improve traffic prediction, and the experimental results are presented in section 4.5.

4.3 Background

In the "Smart City" domain, many studies have been presented on machine learning and statistical-based approaches. Among other factors, these consider air pollution, noise pollution, atmospheric data, and road traffic. For example, Rosenlund et al. [44] did a comparative analysis of different regression models to predict the spatial distribution of road traffic-related air pollution. Zhang et al. [84] provide an analysis about an uptake in health risks when an increase in road traffic is observed. Their study is based on a simulation modeling that estimates the increase in NO_2 concentration, given an increase in road traffic. Po et al. [101] presented their work on the TRAFAIR project related to the effect of

road traffic on air pollution. Lana et al. [48] showed the relationship between road traffic and air pollutants, using regression models to predict air pollution based on road traffic data. Several works incorporated weather conditions as a part of road traffic prediction. Zhang et al. [67] combined Recurrent Neural Networks and Gated Recurrent Units to predict road traffic considering weather conditions. Ryu et al. [102] proposed an approach called the multi-module deep neural network. There network considers different weather conditions to predict road traffic. Similarly, Dunne et al. [103] account for weather conditions in their road traffic prediction using Neurowavelet Models. In a somewhat different vein, some works present the correlation between noise and traffic. For example, Do et al. [104] assessed the increase in noise pollution and its effect on humans, attributing the increase in noise pollution to road traffic. Nourani et al. [105] applied AI-based empirical models to predict noise pollution using road traffic data. They used data from different roads to evaluate the effects on their experimental results. Similarly, Sotiropoulou [106] applied the CRTN (calculation of road traffic noise) model to predict traffic noise. Several approaches have been proposed to predict traffic noise from traffic intensity [107] [108]. However, in this chapter, our goal is different; we aim to usenoise as a general-purpose sensing feature to improve traffic prediction.

All the works cited above show an association of road traffic with air pollution, noise pollution, and atmospheric variables. Another research path is to use pollution, noise or other measured quantities or levels to determine the causal factor(s). Many works utilize these variables, or other similar variables like electric and magnetic profiles, as signatures to identify and classify the cause(s) or for prediction purposes. For example, Fedele et al. [109] used the acoustic signature (noise) to predict cracks on a road surface. Similarly, Nooralahiyan et al. [110] used acoustic signatures to classify vehicles using a Time-Delay Neural Network. Czyzewski et al. [111] used passive acoustic radar and Doppler radar to count the number of passing vehicles on road and to determine their direction. Similarly, Badii et al. [19] considered weather conditions as a signature with which to predict parking spot availability.

These studies all point to the possibility of introducing the notion of measurable signatures in smart cities in order to identify, classify, and predict correlated events. This step would be an enabler for further studies on synthetic sensing.

Our Previous Work:

In machine learning, feature selection is one of the most important steps. Good features are the ones that are highly correlated with the class label [112]. There are lots of studies, such as [113] which show the importance of features' correlation. In our previous work [114], we evaluated different air pollution and atmospheric signatures to determine how correlated

they are to traffic and how they can contribute to achieve a better prediction of traffic flow in Madrid. We based the concept of signature on the fact that a set of vehicles operating for a specific period of time in the same area will produce a very similar quantity of pollutants (imagine 100 cars in a closed environment, they will produce the same amount of pollutants when operating for the same period of time) [114] [52]. We designated as a "traffic signature" the measured concentration of pollutants produced by the cars in a specific location over a pre-determined period. The website Madrid open data offers traffic intensity and pollutant levels measurements. These can be correlated in certain areas where pollution stations are co-located to nearby sensors. We considered the traffic intensity in the proximity of a pollution sensor at specific time periods.

In this experiment, we used air pollutants, including CO, NO, NO_2, and NO_x along with atmosphere variables (wind speed and temperature) as features. Two experiments were conducted: i) Traffic prediction using air pollution and atmospheric variables as features; and ii) Traffic prediction without using air pollution and atmospheric variables. Experimental results showed that the traffic prediction performance improved when using air pollution and atmospheric variables as features (additional detailed correlation analysis and experimental results are available in [114]).

4.4 Steps for creating an LSTM RNN for improving traffic prediction using noise as an additional feature

Traffic intensity can be correlated to other features. The second step of these experiments envisaged the study of the relation between the traffic and the noise intensity as measured in the City of Madrid. We decided to pursue the experiment by identifying traffic sensors close to noise stations and to use the available time series of value to determine, by means of a LSTM RNN. In order to define the LSTM RNN experiment, these steps were considered: first, an initial analysis of the possible correlations between the traffic and the noise, in order to identify the features for the RNN definition; second, the data preparation accordingly to the identified features; third, the definition, implementation and tuning of the Long-Short Term Memory Recurrent Neural Network. these steps are described in this section.

4.4.1 Traffic to Noise Pattern Analysis

Our methodology is based on our previous work [114], in which we used air pollutants and atmospheric variables to improve traffic forecasting using LSTM Recurrent Neural Networks. As a starting option, we considered the possibility of combining traffic, pollution, and noise data in order to have better prediction while reducing the need for accessing and using the data from many traffic sensors. Unfortunately, the Noise sensors and the Pollution

Stations are not co-located and we opted to focus work on the correlation between Noise to Traffic.

The results of the experiment involving pollution signatures were very encouraging and this led to consider the possibility of identifying other signatures and conduct new experiments to relate them to traffic levels. The motivation came from the seminal work done for more confined environments on the so-called synthetic and general purpose-sensing [54], where the sensing infrastructure comprised only general-purpose sensors and the information and relevant data for describing the environment were inferred by means of AI technologies (i.e., synthetic sensing). Traffic is one of the larger sources of noise pollution in urban areas [115] and a few cities are collecting this information (e.g., Madrid, Dublin). Noise can be considered as a general purpose feature. We conducted experiments, presented here, using noise signatures (with the noise levels detected in Madrid by means of a deployed network of noise sensors) and traffic intensity time series to predict traffic flow in Madrid and to figure how if and how this general sensing feature can help in improving the predictive results compared to the baseline case of traffic time series alone. As in the pollution experiments and for an initial verification, we carried out a few correlation analysis analysis to see how aligned traffic flow and noise are. The goal is also to determine the features to be used for the applications of the LSTM RNN. We chose a noise sensor and a traffic intensity sensor situated approximately 20 meters from each other (Figure 4.2). We selected sensors in close proximity to each other for these experiments in order to detect possible patterns between traffic and noise and to be confident that the noise levels were indeed caused by traffic. Figure 4.1 shows the deployed noise sensors in Madrid. Our experiments are based on 3 months traffic and noise data provided by the Madrid City Council [73]. In our previous work on pollution, we used one year data. In this work, we had access to three months hourly noise data from Madrid City Council. Affect of data set size on performance of Neural Network model may vary with respect to application domains. However, there are studies in the literature that deal with applying neural network models on smaller data sets. For example, D'souza et al [116] performed an in-depth study on neural network optimization for small data sets. They showed that neural network optimization can provide high accuracy even on a smaller data set. They conducted their experiments on data sets with different sample sizes (100, 500, and 1000). Lemarchand [117] used only 20 days data with 975 samples as training data and 109 samples as testing data to perform COVID-19 forecast using LSTM RNN. Similarly, there are lots of studies, such as [118] [119] [120] in which few weeks of data were used for traffic forecasting. Comparing our data set size with the data set sizes used in above mentioned studies, we considered three months data sufficient and proceeded with the experimentation. However for consolidating the results, we took technical steps (further discussed in section 4.4.4) to guarantee the solidity of our approach. In addition,

when more noise data will be available from Madrid or other smart cities, we will further process the data.

We plotted three months (January 2019 to March 2019) on an averaged hourly graph using the sensors showed in figure 4.2. The pattern between noise and traffic is shown in figure 4.4. It is clear that except for a very few time instances, the noise is aligned with the traffic pattern: i.e., it increases when the traffic increases and it shows a decreasing pattern when traffic is diminishing. The small misalignment in the pattern at some points will be investigated further in a subsequent study. The hypothesis is that this is due to an increase in noise related to other activities or due to an unusual traffic behavior (e.g., a traffic jam, which we plan to investigate in future work).

4.4.2 Data organization

4.4.2.1 Dataset

To conduct the experiments, traffic and noise data were collected from open data of the city of Madrid [73]. The dataset covered three months (January 2019 to March 2019) of hourly noise and traffic data. Traffic intensity in traffic data represents the number of vehicles/hour, while the noise value was calculated in dBA.

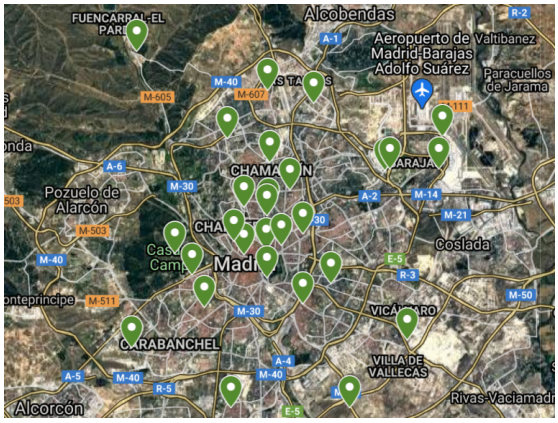


Figure 4.1: Noise sensors deployed in Madrid

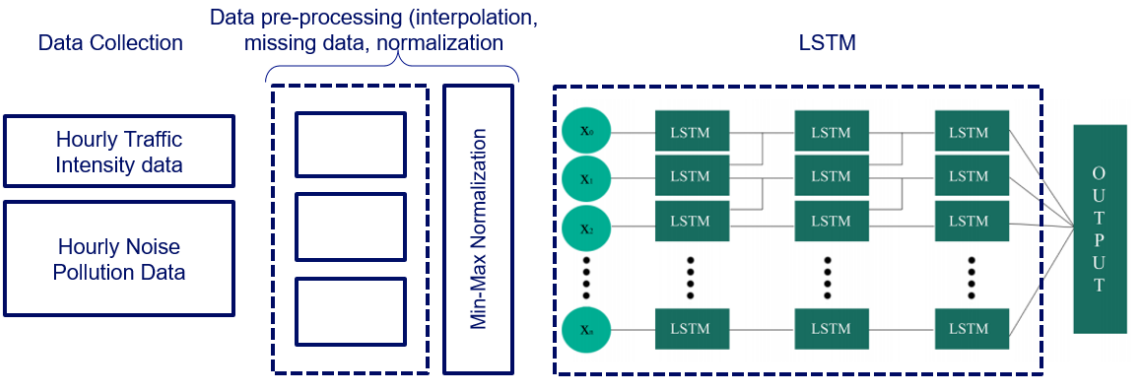


Figure 4.3: Machine Learning Pipeline

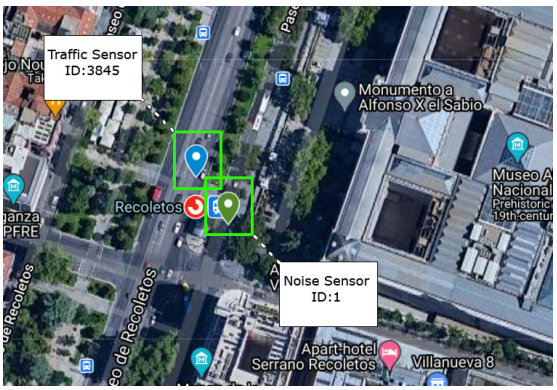


Figure 4.2: Location of the traffic and noise sensors for our study, deployed on/near one of the roads in Madrid's city center

The noise data of Madrid were available at different statistical noise levels (L_{10}, L_{50}, L_{90}). Statistical noise level L_{10} is often used for a traffic noise assessment and for measurements of noise levels due to traffic [121]. A series of experiments along these lines were conducted by the Environmental Protection Department of Hong Kong [122]. Given all these findings, we decided to use the L_{10} statistical noise level for noise data.

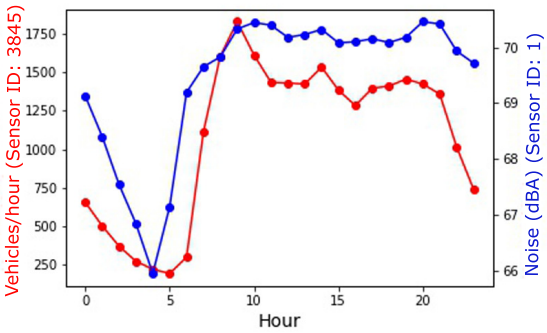


Figure 4.4: Graph of three months' averaged, hourly Traffic-Noise correlation

Statistical Noise Levels

Statistical noise levels, generally represented as L_n, are used to measure environmental noise, for example traffic noise and other ambient noises. These statistical levels help to observe the fluctuating behavior of different noise pollution sources. As mentioned above, there are different statistical levels available, among which, L_{10}, L_{50}, and L_{90} are the ones most commonly used. These statistical levels give information about exceeding percentages. For example, L_{10} presents the noise level that exceeded a base level 10% of the time in a given time interval (noise levels L_n are explained in detail in [123]). A postfix digit with the letter $"L"$ represents the percentage.

Table 4.1 lists the features used to train the LSTM RNN model.

Table 4.1: Features used to train the LSTM RNN model

Feature	**Values/Unit**
Hour	0-23
Week Day	0-6
Noise	dBa
Traffic	Vechiles/hour

4.4.3 Data Pre-Processing

Data are always collected with imperfections. Datasets require pre-processing before they can be used to train machine/deep learning models. Pre-processing is one of the most important steps, as it affects machine learning model accuracy [124]. One of the major challenges with time-series data is data inconsistency. Unfortunately, due to several reasons, time-series data may have problems like timestamps irregularity or removed/missing data points [125]. These problems are often inevitable. We had the same problems in our Madrid

traffic data. In order to deal with that, we applied one of the well-known approaches, called data interpolation. Many studies in interpolation [126] [125] have demonstrated the ability to "calculate" reliable data sets [127] [128]. In our experiments, we applied linear interpolation which tends to search a straight line between two data end points.

Linear interpolation between these two points can be represented as follows:

$$X_i = \frac{X_A - X_B}{a - b}(i - b) + X_b \tag{4.1}$$

where X_A and X_B are the end points, and a, b, and i are the indexes. After pre-processing, data (with features mentioned in table 4.1) were fed to LSTM RNN to train model for road traffic prediction. Figure 4.3 presents the machine learning pipeline for our traffic prediction model.

4.4.4 Long-Short Term Memory Recurrent Neural Network

We used an LSTM RNN to run the experiments for several reasons: it is a well-known deep learning model for time-series prediction; it is very practical for the prediction of time-series with long temporal dependencies [129]; and for consistency with our previous work (we used LSTM RNNs to study the relationship between pollution levels and traffic intensity [114]). Unlike other neural network models, the LSTM RNN has memory cells instead of hidden units. A memory cell (shown in figure 4.5) consists of:

- **An Input gate:** Deals with the input;
- **A Cell state:** To add/remove information;
- **A Forget gate:** Responsible for deciding the fraction of information to keep;
- **An Output gate:** Generates the LSTM output;
- **A Sigmoid layer:** The output generated in the range [0 1] by the sigmoid layer is used to decide if there should be a flow or not; and
- **A Tanh layer:** A vector generated by the Tanh layer is added to the state.

The ability of LSTM RNNs to consider single data points as well as the sequence of data points makes it very useful for time-series data. Architectural details and the working of memory cells in LSTM RNNs are provided in our previous work [114]. We used four hidden layers, with 164, 84, 42, and 21 hidden nodes, respectively. We reached this configuration by using the hit and trail method, following configuration guidelines from [34]. In order to stop the model from going into overfitting [83], we applied early stopping with the patience

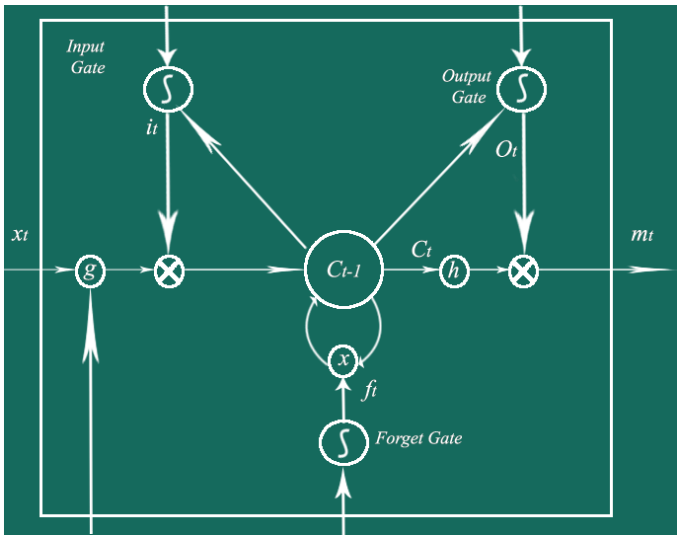


Figure 4.5: Internal architecture of a memory unit of an LSTM RNN)

value 10. To capture the pattern from the previous week, we used a 168 lookback step (168 hours in one week, as in our previous work). Table 4.2 summarizes the hyperparameter values. Next, we applied data normalization, which guarantees the quality of data. Data normalization is required in order to maintain the general data distribution [80]. Our dataset was not having significant outliers. We were more concerned about keeping the exact same scale. Min-Max normalization deals with keeping the exact same scale better than other normalization techniques, such as Z-Score. Therefore, a Min-Max scalar was used to normalize the data. In Min-Max normalization, data is mapped into pre-defined ranges (e.g., [0,1]). The values of each attribute in the data are defined according to their minimum and maximum value which guarantees the quality of data before it is fed to a Machine/deep Learning model [80]. The mathematical representation of a Min-Max scalar can be found in [114]. Following the machine learning modeling convention, we used 67% (≈1448 samples) as training data and 33% (≈ 712 samples) as validation data.

Table 4.2: Hyperparameter Values for LSTM RNN

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Hidden Layers	(168,84,42,21)	Loss	MAE
Activation Function	tanh	Early Stopping (ES)	Enabled
Optimizer	Adam	ES Patience Value	10
Lookback Step	168		

4.5 Experimental Results

In this section, we provide the results of experiments for Traffic prediction using noise pollution as a signature. The mean absolute error (MAE) was used as an evaluation metric. It is represented mathematically below:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_{i,pred}|}{n} \tag{4.2}$$

where $y_i, y_{i,pred}$, and n represent the ground truth, the predicted value against the given ground truth, and the total number of samples, respectively.

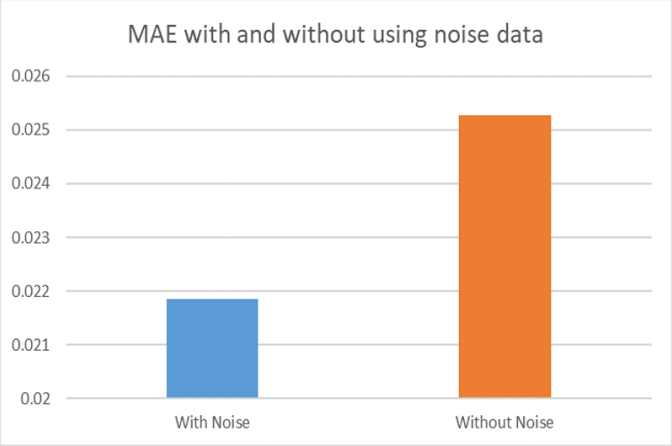


Figure 4.6: Mean absolute error with and without using noise data

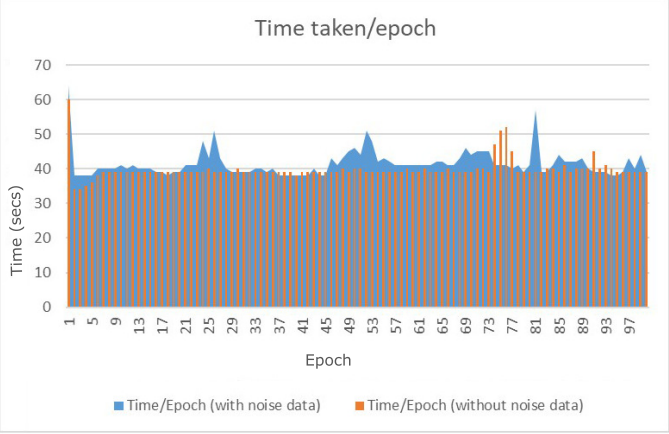


Figure 4.7: Time taken for each epoch while training (in seconds)

Similar to our previous work, we conducted two experiments to assess the approach: i) Using noise data along with traffic time-series data; and 2) Using only traffic time-series

Table 4.3: Time complexity table

	With noise data	**Without noise data**
Average Time/Epoch (seconds)	41.39	39.68
Total Training Time (seconds)	4139	3968
Total Training Time (minutes)	68.98	66.13

data without including noise data (a baseline model utilized to evaluate the effectiveness of the addition of noise data).

As mentioned above, in order to evaluate the effectiveness of the addition of a noise signature, we first trained an LSTM RNN model without noise data, calling it a baseline model. Next, we added noise data and trained the model. The performances of both models were compared in terms of mean absolute error. Figure 4.6 shows that the addition of noise data resulted in improvements in the MAE of 13.48%. We also evaluated both models in terms of time complexity. Figure 4.7 shows that there is no significant difference in the training behavior in terms of time complexity. Table 4.3 further elaborates the model training time complexity, showing that when noise data were used to predict traffic intensity, the model took only 2.85 extra minutes to be trained.

Chapter 5

Conclusion and Future Work

Contents

5.1 Conclusion

Latest advancement in internet, AI, Hardware, and Data technologies have opened the doors for many research directions to make this world a better place to live. Smart city is one of those prominent directions in which a number of work is being done. It is an emerging trend. Given its effectiveness, more and more countries are putting efforts to convert their cities into smart cities. It encapsulates wide range of domains like health, education, environment, and intelligent transportation system (ITS), to name a few. In this thesis, we targeted one of the very popular domains of Smart City, i.e.,ITS. We carried out our research on two of of the most important parts of ITS: i) Smart Parking, and ii) Traffic Forecasting. The summary of our work in terms of contribution is provided below.

5.1.1 Summary and Insights of Contributions

In this section, we provide the summary of each contribution, as well as the insights gained from each contribution.

- **Parking spots availability prediction and recommendation**: In this work, the analysis took into consideration some of the well-known and most used algorithms, newer or emerging ones could be considered and analyzed in further studies. The novelty of the study is related to the compared analysis of them based on data sets of different sizes but containing data reflecting the real environment. Our goal was to find the optimized Machine/Deep Learning model for the prediction of parking space availability by performing comparative analysis of five different well-known Machine/Deep Learning Models: Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and the Voting Classifier/Ensemble Learning (EL) approach. This work presents the numerical results based on K-fold cross-validation. Precision, Recall, F1-score, and Accuracy were used as evaluation metrics. We conducted experiments to predict the availability of parking spots with 10- and 20-min prediction validity, setting 60% and 80% as availability thresholds. These features can be tuned according to the needs of users and the specific experience of the service to provide to users. These values were considered meaningful and useful in an environment such as Santander. One of the main contributions of this chapter is that it seeks to evaluate if a better result can be produced for the parking space availability prediction problem by using less complex algorithms. From the results of our comparative analysis, we found that *Decision Tree* is the optimal solution for the parking space availability prediction problem, and that Ensemble Learning was a close second-best model. With this comparison, we observed that one of the simplest algorithms (KNN) consistently outperformed one of the computationally complex algorithms (Multilayer

Perceptron). We also conducted experiments to observe the effect of training data size on all five of the ML/DL algorithms compared in this chapter.

- **Road traffic prediction improvement using air pollution and atmospheric data:** Traffic forecasting is one of the most important tasks for big cities. Accurate traffic flow forecasting can help drivers to better plan their trips. To provide accurate traffic flow forecasting, this contribution aims to improve the road traffic forecasting. To do so, we combined air pollutants and atmospheric data with traffic intensity data to forecast traffic flow in Madrid, Spain. To evaluate the performance, in the second step, only timestamped traffic intensity data were used to forecast traffic flow, and then those results were compared with the results from the experiments at step one. The comparison was carried out to observe the effect of adding air pollutants and atmospheric data to forecast the traffic flow. We used a long short-term memory recurrent neural network (LSTM RNN) to perform traffic flow forecasting, with time-series traffic flow, air pollution, and atmospheric data collected from the open datasets of Madrid, Spain. Air pollutants (CO, NO, NO_2, NO_X, and O_3), which are associated with road traffic, were considered as the input features, along with atmospheric variables (wind speed, wind direction, temperature and pressure), because in air pollution dispersion models, these features influence the dispersion of air pollution. Together these features helped the model to better forecast the traffic flow. Experimental results show that addition of air pollutant and atmospheric information with timestamp improved the performance.

- **Road traffic forecasting improvement using noise pollution** This work is an extension of our above mentioned contribution. It aims to improve the traffic forecasting by using noise pollution. Much work has been done to show how traffic flow can be used to predict air pollution and noise pollution. In our previous work, contribution C2, we used air pollution and atmospheric levels in specific location and periods, i.e., signatures, to improve traffic prediction in Madrid. Those results proved that considering air pollution levels and atmospheric data helped to improve traffic prediction. Motivated by those results, we investigated another signature type associated with road traffic, i.e., noise pollution. In this work, we discussed how an LSTM RNNs was trained using noise data to produce improved traffic prediction. To assess the effectiveness of adding noise signatures, we compared its trained model performance with a baseline model that was trained using only traffic time-series data without noise data. Our experimental results showed that the addition of noise data improved the performance of LSTM RNN by 13.48%.

5.2 Future Work and Challenges

In this section, we shed a light on some on the future work to extend the work in this thesis.

At first, we performed a comparatively analysis of different machine learning and deep learning approaches to predict individual parking spots on open street areas. This work can be extended to (i) demonstrate the efficiency of the Decision Tree model by integrating it into the smart parking application of Santander, Spain and obtain user feedback, and (ii) use the Santander, Spain road traffic data set and offer recommendations for parking spot management based on traffic data. A recommendation system can integrate the prediction functionality by adopting the algorithm that is better aligned and predicts results with the needed precision. On this basis, additional features and functions can improve the customer experience. Some features can be devoted to improve and simplify the search for an available parking space; however, in conjunction with the government of the city or considering some pollution related considerations, some novel policies for directing people to the "right" destination could be considered, implemented, and verified in the field.

In our second study, we investigated the relationship between road traffic, air pollution, and atmospheric variables in terms of correlation analysis. This work can further be extended to assess the effects of seasons, e.g., summer and winter. Traffic patterns are likely to be different in August in Europe, as many people leave cities and go on vacations. Moreover, we want to identify the percentage of air pollution contributed by road traffic and heating/cooling systems in homes, offices, and factories. In addition, air pollution dispersion models like Ausplume and Calpuss can be considered to better understand the behavior of air pollution. The correlation between air pollution and traffic intensity may differ in different areas of the city. Density of the infrastructure can have an impact on the correlation. In this work, we only considered two areas in Madrid. However, as an extension, multiple areas and their infrastructure can be taken into account to observe the correlation between traffic flow and air pollutants. As a goal, it would be to understand if it is possible to analyze the 'signatures' / traces of pollution to derive and predict information for correlated phenomena. At the same time, satellite pollution measurements can be taken into consideration to understand if they can be used together with ground values to better identify the correlations. In this work, we considered one of the popular neural network models, i.e., LSTM recurrent neural network. However, some studies, such as our previous work [114] show that traditional machine learning models can sometimes perform better than deep learning techniques. In addition to traditional machine learning models, statistical models have also been found to perform better than machine learning models [130]. Hence, it is an open research question to choose the better machine/deep learning model combined with air pollution and atmospheric data.

In addition, in the future, it can also be investigated how to optimize the fusion of different sources of information to improve the prediction for relevant phenomena in the cities. The deployment and maintenance of a large sensor network for traffic and air quality monitoring is a large investment that requires careful planning to be effective and practical. There are a few cities (Madrid is one), that have similar deployment and provide open access to data [73] [131] [132]. Many other cities cannot afford such an investment. This means that monitoring may be very active in certain areas while areas nearby are not similarly controlled. Work can be done on pollution data analysis to verify if it is possible to adequately monitor pollution and to derive and predict phenomena related/associated to it. Another aspect that can be further studied is the possibility offered by the fusion of data in reducing the number of sensors in a city without lowering the information quality, which will ultimately lead to a reduction in cost. For instance, in Madrid, some traffic sensors could be eliminated in favor of more air control sensors if a strong relationship can be verified between traffic and pollution levels.

As a third contribution, we studied the relationship between noise pollution and traffic. This contribution is an initial work to determine the effectiveness of a signature-based approach to improve traffic prediction in smart cities. A long-term goal of our work can be to investigate general-purpose sensing to reduce the number of dedicated sensor networks and to create the conditions for inferring data by means of synthetic sensing, data fusion, and related AI techniques. In the future, work on the following challenges can be done:

- Transfer learning using signature data (noise and pollution): Models can be trained a using signatures and traffic data in one area and then use the trained model to predict traffic intensity in another area.

- Combining signatures (e.g., noise data, air pollution data, and atmospheric data) together to improve the traffic prediction.

- Develop a way to create and validate data sets in non-sensored areas by exploiting interpolation, models and signatures from similar areas to provide effective predictions. "Ad hoc" experiments can be carried out to validate the results.

- In order to achieve synthetic sensing (the usage of one or more types of sensing capabilities to provide the sensing which requires special and dedicated sensing capabilities), identify a basic number of sensing capabilities (e.g., vibration, noise, pollution. 3-D shaping, magnetic data and others) that are sufficient to adequately measure the largest possible number of phenomena in open complex environments. For example, developing noise and air pollution signatures with respect to road traffic, and then using those signatures to predict the road traffic.

- Some studies show that weather conditions and the physical shape of an environment affect the auditory space [133]. Therefore, environmental structures and weather conditions can be considered in sensing systems.

This research path seems to be very promising. If it proves successful, it may have a real impact on how the infrastructure of smart cities is designed and implemented and how services can be offered to citizens. Most importantly, it will allow complexity to be moved from specialized sensor networks to general-purpose ones thereby promote the development of software infrastructure capable of exploiting the newer AI technologies.

References

[1] Smart cities. Last Accessed: February 26, 2022.

[2] Paul Melnyk, Soufiene Djahel, and Farid Nait-Abdesselam. Towards a smart parking management system for smart cities. In *2019 IEEE International Smart Cities Conference (ISC2)*, pages 542–546. IEEE, 2019.

[3] Ammar Haydari and Yasin Yilmaz. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[4] Li Zhu, Fei Richard Yu, Yige Wang, Bin Ning, and Tao Tang. Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):383–398, 2018.

[5] Juan Guerrero-Ibáñez, Sherali Zeadally, and Juan Contreras-Castillo. Sensor technologies for intelligent transportation systems. *Sensors*, 18(4):1212, 2018.

[6] Ibm survey. Accessed: 20-08-2019.

[7] Ivan Klandev, Marta Tolevska, Kostadin Mishev, and Dimitar Trajanov. Parking availability prediction using traffic data services. 2020.

[8] Pablo Martín Calvo, Bas Schotten, and Elenna R Dugundji. Assessing the predictive value of traffic count data in the imputation of on-street parking occupancy in amsterdam. *Transportation research record*, 2675(12):330–341, 2021.

[9] Ali Ziat, Bertrand Leroy, Nicolas Baskiotis, and Ludovic Denoyer. Joint prediction of road-traffic and parking occupancy over a city with representation learning. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 725–730. IEEE, 2016.

[10] Andrew Koster, Allysson Oliveira, Orlando Volpato, Viviane Delvequio, and Fernando Koch. Recognition and recommendation of parking places. In *Ibero-American Conference on Artificial Intelligence*, pages 675–685. Springer, 2014.

[11] Wan-Joo Park, Byung-Sung Kim, Dong-Eun Seo, Dong-Suk Kim, and Kwae-Hi Lee. Parking space detection using ultrasonic sensor in parking assistance system. In *2008 IEEE intelligent vehicles symposium*, pages 1039–1044. IEEE, 2008.

[12] Mikko Rinne, Seppo Törmä, and D Kratinov. Mobile crowdsensing of parking space using geofencing and activity recognition. In *10th ITS European Congress, Helsinki, Finland*, pages 16–19, 2014.

[13] Muhammed Abdurrahman Hazar, Niyazi Odabasioglu, Tolga Ensari, Yusuf Kavurucu, and OF Sayan. Performance analysis and improvement of machine learning algorithms for automatic modulation recognition over rayleigh fading channels. *Neural Computing and Applications*, 29(9):351–360, 2018.

[14] Barath Narayanan Narayanan, Ouboti Djaneye-Boundjou, and Temesguen M Kebede. Performance analysis of machine learning and pattern recognition algorithms for malware classification. In *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)*, pages 338–342. IEEE, 2016.

[15] Rosamaria Elisa Barone, Tullio Giuffrè, Sabato Marco Siniscalchi, Maria Antonietta Morgano, and Giovanni Tesoriere. Architecture for parking management in smart cities. *IET Intelligent Transport Systems*, 8(5):445–452, 2014.

[16] Jihoon Yang, Jorge Portilla, and Teresa Riesgo. Smart parking service based on wireless sensor networks. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pages 6029–6034. IEEE, 2012.

[17] Shi Dong, Mingsong Chen, Lei Peng, and Huiyun Li. Parking rank: A novel method of parking lots sorting and recommendation based on public information. In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pages 1381–1386. IEEE, 2018.

[18] Eleni I Vlahogianni, Konstantinos Kepaptsoglou, Vassileios Tsetsos, and Matthew G Karlaftis. A real-time parking prediction system for smart cities. *Journal of Intelligent Transportation Systems*, 20(2):192–204, 2016.

[19] Claudio Badii, Paolo Nesi, and Irene Paoli. Predicting available parking slots on critical and regular services by exploiting a range of open data. *IEEE Access*, 6:44059–44071, 2018.

[20] Yanxu Zheng, Sutharshan Rajasegarar, and Christopher Leckie. Parking availability prediction for sensor-enabled car parks in smart cities. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6. IEEE, 2015.

[21] Andrés Camero, Jamal Toutouh, Daniel H Stolfi, and Enrique Alba. Evolutionary deep learning for car park occupancy prediction in smart cities. In *International Conference on Learning and Intelligent Optimization*, pages 386–401. Springer, 2018.

[22] Fengquan Yu, Jianhua Guo, Xiaobo Zhu, and Guogang Shi. Real time prediction of unoccupied parking space using time series model. In *2015 International Conference on Transportation Information and Safety (ICTIS)*, pages 370–374. IEEE, 2015.

[23] Nazia Bibi, Muhammad Nadeem Majid, Hassan Dawood, and Ping Guo. Automatic parking space detection system. In *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*, pages 11–15. IEEE, 2017.

[24] Paula Tătulea, Florina Călin, Remus Brad, Lucian Brâncovean, and Mircea Greavu. An image feature-based method for parking lot occupancy. *Future Internet*, 11(8):169, 2019.

[25] Kou-Yuan Huang, Kai-Ju Chen, Ming-Che Huang, and Liang-Chi Shen. Multilayer perceptron with particle swarm optimization for well log data inversion. In *2012 IEEE International Geoscience and Remote Sensing Symposium*, pages 6103–6106. IEEE, 2012.

[26] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.

[27] Mian Mian Lau and King Hann Lim. Investigation of activation functions in deep belief network. In *2017 2nd international conference on control and robotics engineering (ICCRE)*, pages 201–206. IEEE, 2017.

[28] Archana Singh, Avantika Yadav, and Ajay Rana. K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10), 2013.

[29] Richa Sharma, Aniruddha Ghosh, and PK Joshi. Decision tree approach for classification of remotely sensed satellite data using open source support. *Journal of Earth System Science*, 122(5):1237–1247, 2013.

[30] Santander facility, smart santander. Accessed: 20-12-2021.

[31] Worldwide interoperability for semantics iot (wise-iot), h2020 eu-kr project. Accessed: 18-03-2019.

[32] Parking sensors at santander, spain. Accessed: 18-03-2019.

[33] Scikit learn library. Accessed: 25-09-2019.

[34] Jeff Heaton. Aifh, volume 3: deep learning and neural networks. *Journal of Chemical Information and Modeling*, 3, 2015.

[35] Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize f1 measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–239. Springer, 2014.

[36] Ganda Yoga Swara et al. Implementation of haversine formula and best first search method in searching of tsunami evacuation route. In *IOP Conference Series: Earth and Environmental Science*, volume 97, page 012004. IOP Publishing, 2017.

[37] JM Schmidt, O Tendwa, and MM Bruwer. Traffic impact of the its time event. In *37th Annual Southern African Transport Conference*, page 704. Jukwaa Media, 2018.

[38] Yan Kuang, Barbara TH Yen, Emiliya Suprun, and Oz Sahin. A soft traffic management approach for achieving environmentally sustainable and economically viable outcomes: An australian case study. *Journal of environmental management*, 237:379–386, 2019.

[39] Toon Bogaerts, Antonio D Masegosa, Juan S Angarita-Zapata, Enrique Onieva, and Peter Hellinckx. A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data. *Transportation Research Part C: Emerging Technologies*, 112:62–77, 2020.

[40] Lazar Lazić, Mira Aničić Urošević, Zoran Mijić, Gordana Vuković, and Luka Ilić. Traffic contribution to air pollution in urban street canyons: Integrated application of the ospm, moss biomonitoring and spectral analysis. *Atmospheric Environment*, 141:347–360, 2016.

[41] A report by who. Last Accessed: March 27, 2020.

[42] European commission. analizing traffic flows in madrid city. Last Accessed: March 23, 2020.

[43] Piotr S Maciąg, Nikola Kasabov, Marzena Kryszkiewicz, and Robert Bembenik. Air pollution prediction with clustering-based ensemble of evolving spiking neural networks and a case study for london area. *Environmental Modelling & Software*, 118:262–280, 2019.

[44] Mats Rosenlund, Francesco Forastiere, Massimo Stafoggia, Daniela Porta, Mara Perucci, Andrea Ranzi, Fabio Nussio, and Carlo A Perucci. Comparison of regression models with land-use and emissions data to predict the spatial distribution of traffic-related air pollution in rome. *Journal of exposure science & environmental epidemiology*, 18(2):192–199, 2008.

[45] Dan L Crouse, Mark S Goldberg, and Nancy A Ross. A prediction-based approach to modelling temporal and spatial variability of traffic-related air pollution in montreal, canada. *Atmospheric environment*, 43(32):5075–5084, 2009.

[46] Stuart Batterman, Rajiv Ganguly, and Paul Harbin. High resolution spatial and temporal mapping of traffic-related air pollutants. *International journal of environmental research and public health*, 12(4):3646–3666, 2015.

[47] Hai-Bang Ly, Lu Minh Le, Luong Van Phi, Viet-Hung Phan, Van Quan Tran, Binh Thai Pham, Tien-Thinh Le, and Sybil Derrible. Development of an ai model to measure traffic air pollution from multisensor and weather data. *Sensors*, 19(22):4941, 2019.

[48] Ibai Lana, Javier Del Ser, Ales Padró, Manuel Vélez, and Carlos Casanova-Mateo. The role of local urban traffic and meteorological conditions in air pollution: A data-based case study in madrid, spain. *Atmospheric Environment*, 145:424–438, 2016.

[49] Ana Russo, Pedro G Lind, Frank Raischel, Ricardo Trigo, and Manuel Mendes. Neural network forecast of daily pollution concentration using optimal meteorological data at synoptic and local scales. *Atmospheric Pollution Research*, 6(3):540–549, 2015.

[50] Andrea Brunello, Joanna Kamińska, Enrico Marzano, Angelo Montanari, Guido Sciavicco, and Tomasz Turek. Assessing the role of temporal information in modelling short-term air pollution effects based on traffic and meteorological conditions: a case study in wrocław. In *European Conference on Advances in Databases and Information Systems*, pages 463–474. Springer, 2019.

[51] World economic forum, this is why people live, work, and stay in a growing city. Last Accessed: March 27, 2020.

[52] Pallavi Pant, Zongbo Shi, Francis D Pope, Roy M Harrison, et al. Characterization of traffic-related particulate matter emissions in a road tunnel in birmingham, uk: Trace metals and organic molecular markers. *Aerosol and Air Quality Research*, 17(1):117–130, 2017.

[53] Xueying Zhang, Elena Craft, and Kai Zhang. Characterizing spatial variability of air pollution from vehicle traffic around the houston ship channel area. *Atmospheric Environment*, 161:167–175, 2017.

[54] Gierad Laput, Yang Zhang, and Chris Harrison. Synthetic sensors: Towards general-purpose sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3986–3999, 2017.

[55] Tian Guo, Zhao Xu, Xin Yao, Haifeng Chen, Karl Aberer, and Koichi Funaya. Robust online time series prediction with recurrent neural networks. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 816–825. Ieee, 2016.

[56] Byoungsuk Ji and Ellen J Hong. Deep-learning-based real-time road traffic prediction using long-term evolution access data. *Sensors*, 19(23):5327, 2019.

[57] Wangyang Wei, Honghai Wu, and Huadong Ma. An autoencoder and lstm-based traffic flow prediction method. *Sensors*, 19(13):2946, 2019.

[58] Yaguang Li and Cyrus Shahabi. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *Sigspatial Special*, 10(1):3–9, 2018.

[59] Roozbeh Ketabi, Mimonah Al-Qathrady, Babak Alipour, and Ahmed Helmy. Vehicular traffic density forecasting through the eyes of traffic cameras; a spatio-temporal machine learning study. In *Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pages 81–88, 2019.

[60] Dongjie Zhu, Haiwen Du, Yundong Sun, and Ning Cao. Research on path planning model based on short-term traffic flow prediction in intelligent transportation system. *Sensors*, 18(12):4275, 2018.

[61] Qinzhong Hou, Junqiang Leng, Guosheng Ma, Weiyi Liu, and Yuxing Cheng. An adaptive hybrid model for short-term urban traffic flow prediction. *Physica A: Statistical Mechanics and its Applications*, 527:121065, 2019.

[62] Jinjun Tang, Xinqiang Chen, Zheng Hu, Fang Zong, Chunyang Han, and Leixiao Li. Traffic flow prediction based on combination of support vector machine and data denoising schemes. *Physica A: Statistical Mechanics and its Applications*, 534:120642, 2019.

[63] Wei Wang, Hanyu Zhang, Tong Li, Jianhua Guo, Wei Huang, Yun Wei, and Jinde Cao. An interpretable model for short term traffic flow prediction. *Mathematics and Computers in Simulation*, 171:264–278, 2020.

[64] Yalda Rajabzadeh, Amir Hossein Rezaie, and Hamidreza Amindavar. Short-term traffic flow prediction using time-varying vasicek model. *Transportation Research Part C: Emerging Technologies*, 74:168–181, 2017.

[65] Shidrokh Goudarzi, Mohd Nazri Kama, Mohammad Hossein Anisi, Seyed Ahmad Soleymani, and Faiyaz Doctor. Self-organizing traffic flow prediction with an optimized deep belief network for internet of vehicles. *Sensors*, 18(10):3459, 2018.

[66] Afshin Abadi, Tooraj Rajabioun, and Petros A Ioannou. Traffic flow prediction for road transportation networks with limited traffic data. *IEEE transactions on intelligent transportation systems*, 16(2):653–662, 2014.

[67] Da Zhang and Mansur R Kabuka. Combining weather condition data to predict traffic flow: a gru-based deep learning approach. *IET Intelligent Transport Systems*, 12(7):578–585, 2018.

[68] Analyzing traffic flows in madrid city. Last Accessed: June 23, 2020.

[69] Analyzing traffic flows in madrid city. Last Accessed: June 23, 2020.

[70] Lykourgos Tsirigotis, Eleni I Vlahogianni, and Matthew G Karlaftis. Does information on weather affect the performance of short-term traffic forecasting models? *International journal of intelligent transportation systems research*, 10(1):1–10, 2012.

[71] Xiujuan Xu, Benzhe Su, Xiaowei Zhao, Zhenzhen Xu, and Quan Z Sheng. Effective traffic flow forecasting using taxi and weather data. In *International Conference on Advanced Data Mining and Applications*, pages 507–519. Springer, 2016.

[72] European commission directorate-general for the environment. Last Accessed: May 07, 2020.

[73] Open data portal of the madrid city council. Last Accessed: Feb 02, 2020.

[74] R Baldauf, N Watkins, D Heist, C Bailey, P Rowley, and R Shores. Near-road air quality monitoring: factors affecting network design and interpretation of data. *Air Quality, Atmosphere & Health*, 2(1):1–9, 2009.

[75] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[76] Linchao Li, Jian Zhang, Yonggang Wang, and Bin Ran. Missing value imputation for traffic-related time series data based on a multi-view learning method. *IEEE Transactions on Intelligent Transportation Systems*, 20(8):2933–2943, 2018.

[77] Koredianto Usman and Mohammad Ramdhani. Comparison of classical interpolation methods and compressive sensing for missing data reconstruction. In *2019 IEEE International Conference on Signals and Systems (ICSigSys)*, pages 29–33. IEEE, 2019.

[78] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.

[79] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.

[80] SC Nayak, Bijan B Misra, and Himansu Sekhar Behera. Impact of data normalization on stock index forecasting. *International Journal of Computer Information Systems and Industrial Management Applications*, 6(2014):257–269, 2014.

[81] Vatsal Gajera, Rishabh Gupta, Prasanta K Jana, et al. An effective multi-objective task scheduling algorithm using min-max normalization in cloud computing. In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 812–816. IEEE, 2016.

[82] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[83] Lutz Prechelt. Neural networks: Tricks of the trade. *Lecture Notes in Computer Science*, 1524:53–67, 1998.

[84] Lun Zhang, Qiuchen Liu, Wenchen Yang, Nai Wei, and Decun Dong. An improved k-nearest neighbor model for short-term traffic flow prediction. *Procedia-Social and Behavioral Sciences*, 96:653–662, 2013.

[85] Li Li, Xiaonan Su, Yanwei Wang, Yuetong Lin, Zhiheng Li, and Yuebiao Li. Robust causal dependence mining in big data network and its application to traffic flow predictions. *Transportation Research Part C: Emerging Technologies*, 58:292–307, 2015.

[86] Claudia Perlich, Foster Provost, and Jeffrey Simonoff. Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research*, 2003.

[87] Claudia Perlich. Ibm research report: Learning curves in machine learning. Last Accessed: May 10, 2020.

[88] Vikas Nimesh, Debojit Sharma, V Mahendra Reddy, and Arkopal Kishore Goswami. Implication viability assessment of shift to electric vehicles for present power generation scenario of india. *Energy*, 195:116976, 2020.

[89] Tuba Bakıcı, Esteve Almirall, and Jonathan Wareham. A smart city initiative: the case of barcelona. *Journal of the knowledge economy*, 4(2):135–148, 2013.

[90] Paolo Neirotti, Alberto De Marco, Anna Corinna Cagliano, Giulio Mangano, and Francesco Scorrano. Current trends in smart city initiatives: Some stylised facts. *Cities*, 38:25–36, 2014.

[91] Rong Du, Paolo Santi, Ming Xiao, Athanasios V Vasilakos, and Carlo Fischione. The sensable city: A survey on the deployment and management for smart city monitoring. *IEEE Communications Surveys & Tutorials*, 21(2):1533–1560, 2018.

[92] Pablo Sotres, Juan Ramón Santana, Luis Sánchez, Jorge Lanza, and Luis Muñoz. Practical lessons from the deployment and management of a smart city internet-of-things infrastructure: The smartsantander testbed case. *IEEE Access*, 5:14309–14322, 2017.

[93] Kamila Turečková and Jan Nevima. The cost benefit analysis for the concept of a smart city: How to measure the efficiency of smart solutions? *Sustainability*, 12(7):2663, 2020.

[94] Hadi Habibzadeh, Zhou Qin, Tolga Soyata, and Burak Kantarci. Large-scale distributed dedicated-and non-dedicated smart city sensing systems. *IEEE Sensors Journal*, 17(23):7649–7658, 2017.

[95] Gierad Laput and Chris Harrison. Exploring the efficacy of sparse, general-purpose sensor constellations for wide-area activity sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2):1–19, 2019.

[96] Smart Dublin, Open Data Store. `https://data.smartdublin.ie/dataset`. [Online; accessed 09-December-2020].

[97] Open data portal of the santander city. Last Accessed: March 25, 2020.

[98] Ha-Young Kwak, Joonho Ko, Seungho Lee, and Chang-Hyeon Joh. Identifying the correlation between rainfall, traffic flow performance and air pollution concentration in seoul using a path analysis. *Transportation research procedia*, 25:3552–3563, 2017.

[99] Yongna Jia and Jianwei Ma. What can machine learning do for seismic data processing? an interpolation application. *Geophysics*, 82(3):V163–V177, 2017.

[100] Roberto Minerva, Faraz Malik Awan, and Noel Crespi. Exploiting digital twin as enablers for synthetic sensing. *IEEE Internet Computing*, 2021.

[101] Laura Po, Federica Rollo, Jose Ramon Rios Viqueira, Raquel Trillo Lado, Alessandro Bigi, Javier Cacheiro Lopez, Michela Paolucci, and Paolo Nesi. Trafair: understanding traffic flow to improve air quality. In *2019 IEEE International Smart Cities Conference (ISC2)*, pages 36–43. IEEE, 2019.

[102] Seungyo Ryu, Dongseung Kim, and Joongheon Kim. Weather-aware long-range traffic forecast using multi-module deep neural network. *Applied Sciences*, 10(6):1938, 2020.

[103] Stephen Dunne and Bidisha Ghosh. Weather adaptive traffic prediction using neurowavelet models. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):370–379, 2013.

[104] Eriberto Oliveira do Nascimento, Felipe Luz de Oliveira, Lucas Nonato de Oliveira, and Paulo Henrique Trombetta Zannin. Noise prediction based on acoustic maps and vehicle fleet composition. *Applied Acoustics*, 174:107803, 2021.

[105] Vahid Nourani, Hüseyin Gökçekuş, Ibrahim Khalil Umar, and Hessam Najafi. An emotional artificial neural network for prediction of vehicular traffic noise. *Science of the Total Environment*, 707:136134, 2020.

[106] Alexandra Sotiropoulou, Ioannis Karagiannis, Emmanouil Vougioukas, Athanassios Ballis, and Aspasia Bouki. Measurements and prediction of road traffic noise along high-rise building façades in athens. *Noise Mapping*, 7(1):1–13, 2020.

[107] Daljeet Singh, SP Nigam, VP Agrawal, and Maneek Kumar. Vehicular traffic noise prediction using soft computing approach. *Journal of environmental management*, 183:59–66, 2016.

[108] Ahmed Abdulkareem Ahmed and Biswajeet Pradhan. Vehicular traffic noise prediction and propagation modelling using neural networks and geospatial information system. *Environmental monitoring and assessment*, 191(3):190, 2019.

[109] Rosario Fedele, Filippo Giammaria Praticò, and Gianfranco Pellicano. The prediction of road cracks through acoustic signature: Extended finite element modeling and experiments. *Journal of Testing and Evaluation*, 49(4), 2021.

[110] AY Nooralahiyan, Howard R Kirby, and D McKeown. Vehicle classification by acoustic signature. *Mathematical and Computer Modelling*, 27(9-11):205–214, 1998.

[111] Andrzej Czyżewski, Józef Kotus, and Grzegorz Szwoch. Estimating traffic intensity employing passive acoustic radar and enhanced microwave doppler radar sensor. *Remote Sensing*, 12(1):110, 2020.

[112] Mark Andrew Hall. Correlation-based feature selection for machine learning. 1999.

[113] Santiago Egea, Albert Rego Mañez, Belén Carro, Antonio Sánchez-Esguevillas, and Jaime Lloret. Intelligent iot traffic classification using novel search strategy for fast-based-correlation feature selection in industrial environments. *IEEE Internet of Things Journal*, 5(3):1616–1624, 2017.

[114] Faraz Malik Awan, Yasir Saleem, Roberto Minerva, and Noel Crespi. A comparative analysis of machine/deep learning models for parking space availability prediction. *Sensors*, 20(1):322, 2020.

[115] Clara G Sears, Joseph M Braun, Patrick H Ryan, Yingying Xu, Erika F Werner, Bruce P Lanphear, and Gregory A Wellenius. The association of traffic-related air and noise pollution with maternal blood pressure and hypertensive disorders of pregnancy in the home study cohort. *Environment international*, 121:574–581, 2018.

[116] Rhett N D'souza, Po-Yao Huang, and Fang-Cheng Yeh. Structural analysis and optimization of convolutional neural networks with a small sample size. *Scientific reports*, 10(1):1–13, 2020.

[117] Francois Lemarchand. Covid-19 Forecasting with an RNN. `https://www.kaggle.com/frlemarchand/covid-19-forecasting-with-an-rnn`. [Online; accessed 02-June-2021].

[118] Junwei Gao, Ziwen Leng, Yong Qin, Zengtao Ma, and Xin Liu. Short-term traffic flow forecasting model based on wavelet neural network. In *2013 25th Chinese Control and Decision Conference (CCDC)*, pages 5081–5084, 2013.

[119] Pinlong Cai, Yunpeng Wang, and Guangquan Lu. Tunable and transferable rbf model for short-term traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 20(11):4134–4144, 2019.

[120] Kit Yan Chan, Tharam S. Dillon, Jaipal Singh, and Elizabeth Chang. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg–marquardt algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):644–654, 2012.

[121] M Burgess. Relationship between l10 and leq for noise from road traffic. *Australian Road Research*, 8(3), 1978.

[122] Environmental Protection Department, The government of the Hong Kong special administrative region. https://www.epd.gov.hk/epd/noise_education/web/ENG_EPD_HTML/m2/types_3.html. [Online; accessed 23-December-2020].

[123] NTi Audio. https://www.nti-audio.com/en/support/know-how/how-are-percentile-statistics-measured. [Online; accessed 24-December-2020].

[124] Priyanga Chandrasekar, Kai Qian, Hossain Shahriar, and Prabir Bhattacharya. Improving the prediction accuracy of decision tree mining with data preprocessing. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 481–484. IEEE, 2017.

[125] Mathieu Lepot, Jean-Baptiste Aubin, and François HLR Clemens. Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment. *Water*, 9(10):796, 2017.

[126] Sara Mandelli, Federico Borra, Vincenzo Lipari, Paolo Bestagini, Augusto Sarti, and Stefano Tubaro. Seismic data interpolation through convolutional autoencoder. In *SEG Technical Program Expanded Abstracts 2018*, pages 4101–4105. Society of Exploration Geophysicists, 2018.

[127] A Di Piazza, F Lo Conti, Leonardo V Noto, Francesco Viola, and G La Loggia. Comparative analysis of different techniques for spatial interpolation of rainfall data to create a serially complete monthly time series of precipitation for sicily, italy. *International Journal of Applied Earth Observation and Geoinformation*, 13(3):396–408, 2011.

[128] Diego Mendez, Miguel Labrador, and Kandethody Ramachandran. Data interpolation for participatory sensing systems. *Pervasive and Mobile Computing*, 9(1):132–148, 2013.

[129] Danqing Kang, Yisheng Lv, and Yuan-yuan Chen. Short-term traffic flow prediction with lstm recurrent neural network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[130] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889, 2018.

[131] Open data portal of the barcelona city. Last Accessed: March 25, 2020.

[132] Open data portal of the turin city. Last Accessed: March 25, 2020.

[133] Dietmar Offenhuber, Sam Auinger, Susanne Seitinger, and Remco Muijs. Los angeles noise array—planning and design lessons from a noise sensing network. *Environment and Planning B: Urban Analytics and City Science*, 47(4):609–625, 2020.

List of figures

List of tables

Appendix A

Appendix

A.1 Smart City Datasets

We use 2 publicly available sources provided by research community as follows:

Parking Dataset This category consists of the parking dataset of 400 parking sensors deployed in city of Santander, Spain. These data were collected over the period of 9 months. The dataset was constructed as part of H2020 project, the WISE-IoT.

Traffic Intensity Dataset This dataset contains the time-series data from around 4000 traffic intensity sensors deployed in the city of Madrid. This dataset features unique ID of the sensors, timestamp, and traffic intensity with 15 minutes frequency. Open data portal, provided by Madrid City Council, was used to collect this data.

Atmospheric Data This dataset features different atmospheric entities, including temperature, wind speed, wind direction, pressure, and humidity, and time-stamp and sensors' ID. Data is collected with one hour frequency from around 26 weather stations deployed in the city of Madrid. Open data portal, provided by Madrid City Council, was used to collect this data.

Air Pollution Data Air pollution data was collected from the 26 air pollution monitoring stations deployed in the city of Madrid. These data are available with one hour frequency and feature different air pollutants, including $SO, CO, NO, NO_2, O_3, PM_{2.5}, PM_{10}$ with time-stamp and sensors' ID. Open data portal, provided by Madrid City Council, was used to collect this data. 1 year of Air pollution and atmospheric data (January 2019, December 2019) were collected.

Noise Pollution Data The noise pollution data features different statistical noise levels, including L_{10}, L_{50} and L_{90}. Noise data is available on Open Data Portal of Madrid with around 6 hours frequency. However, we requested Madrid City Council to provide us data with one hour frequency. Three months (January 2019 to March 2019)of hourly noise pollution data, featuring different noise level and time-stamp, and noise pollution sensors' ID, were provided.