Doctoral Dissertation.
Interuniversity Doctoral Program in
Urban and Regional Development (36th Cycle)

From Data to Decisions: A Comparative Study of Machine Learning and Statistical Approaches in Estimating Passenger Counts

Ximena R. GARZON RUIZ

Supervisor(s):
Prof. Cristina Pronello, Supervisor

Doctoral Examination Committee:
Prof. Shlomo Bekhor, Technion - Israel Institute of Technology
Prof. Yannis Tyrinopoulos, University of West Attika
Prof. Jesus Gonzalez-Feliu, CERIIM - Research Center in Managerial Intelligence and Innovation
Prof. Dominique Mignot, Université Gustave Eiffel
Prof. Roberto Fontana, Politecnico di Torino

Politecnico di Torino - Università degli Studi di Torino
June 30, 2024

This thesis is licensed under a Creative Commons License, Attribution - Noncommercial - NoDerivative Works 4.0 International: see www.creativecommons.org. The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

I hereby declare that the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Ximena R. Garzón Ruiz Turin, June 30, 2024

Summary

Urban transport planning is inherently complex, requiring accurate representation of transport demand and mobility patterns across spatial and temporal dimensions. Intelligent Transport Systems (ITS) have emerged as pivotal in enhancing the efficiency and environmental sustainability of urban transport networks by automating data collection and processing. Modern technologies, such as mobile applications, Geographic Information Systems (GIS), Internet of Things (IoT) devices, and 5G connectivity, have significantly improved data quality and quantity, outperforming traditional methods like telephone surveys. The transport sector has used data collection methods such as passenger counting (APC) systems to estimate mobility patterns and improve public transport services. Numerous studies in literature have sought to refine the data collected through these systems, aiming to provide a more precise understanding of mobility trends.

While existing literature has focused on enhancing univariate data for prediction and categorisation, efforts have also been made to integrate external variables like environmental and meteorological data to improve prediction accuracy. However, despite these advancements, selecting appropriate systems or processes remains challenging due to the lack of standardised protocols, often resulting in sub-optimal implementations. Moreover, determining the best method for specific datasets is still difficult, as few studies have compared these models using real operational data or considered factors like dataset size and computational power.

This research addresses these gaps by empirically evaluating passenger count estimation models using both artificial intelligence (AI) and traditional statistical techniques, with a focus on data fusion, using real operation data. It also examines the discrepancies between claimed and actual performance of Automated Passenger Counting (APC) systems, which often report accuracy rates of 98% to 99%, yet real-world performance varies widely between 50% and 78%. Specific objectives include developing and implementing diverse passenger count estimation models, applying methods from other fields to the transport sector, evaluating model performance through comprehensive metrics, and providing a comparative analysis to identify each model's strengths and weaknesses in passenger count estimation.

An eight-step methodology was employed to achieve these objectives, involving data collection, processing, and the application of multiple data fusion models. The study tested different models using a data fusion approach that integrated multiple data sources, including two passenger count systems (one camera-based and one Wi-Fi-based), Points of Interest (POIs), meteorological data, pollution data and 57 days of ground truth data collected manually by the author during operation times in Turin. Statistical models were compared against machine learning and deep learning models, with results evaluated using a decision matrix approach based on various factors and potential transport companies' requirements, to then rank the different models.

The research found that LightGBM was the best-performing model, followed by Bayesian regression and XGBoost, based on accuracy, error and scalability metrics across different experiments. Bayesian regression showed strong performance even with smaller datasets, while SARIMAX and Prophet models benefited from data augmentation. Advanced time series models like LSTM and Prophet were less effective, underscoring the importance of aligning algorithm choice with dataset characteristics such as size and seasonality.

The study demonstrates that high-quality, well-distributed data is crucial, and ensemble machine learning algorithms hold significant promise for data fusion in the transport sector. The findings suggest that further research should focus on refining data collection and cleaning methodologies to enhance model performance and exploring time-series model for vehicle load estimation. The research contributes to both the academic field and practical applications by providing insights into reality representation and offering a framework for transport companies to select suitable models for passenger load estimation.

In conclusion, this research provides a thorough evaluation of various data fusion models and offers practical guidelines for their application in the transport sector. The results underscore the importance of data quality and strategic algorithm selection to optimise transport system performance. Further studies should continue to explore advanced data analytics and to select the most effective (typology and size) data sets to address the evolving challenges of urban mobility and to evaluate the reality representation of these models in practical implementations within the field.

Acknowledgment

Words cannot express my gratitude to my supervisor Prof. Cristina Pronello for her invaluable supervision, support and tutelage during the course of my PhD degree. Special thanks are extended to Deepan Anbarasan and Carlos Garcia for their technical support and proofreading my study, and to the TRIS research group, particularly Dr. Pinky Kumawat and Daniel Sebastia, for their immense support during the data collection phase. Their help made it possible to endure the long hours spent on the bus counting passengers Additionally, this endeavour would not have been possible without the support of Leonardo S.p.A. who financed my research, and to GTT and Mobyforall for their collaboration and for enabling me to work with cutting-edge technologies on their vehicles. Finally, I would like to express my gratitude to my parents, my sisters, my husband and my friends. Without their tremendous understanding and encouragement in the past few years, it would have been impossible for me to complete my study.

Dedication

To my wonderful family, especially my parents, who always knew I was up to something good even when I was just dancing around. To my sisters, who kept me grounded and occasionally distracted with their antics.

To my husband, the unofficial proofreader and snack/tea provider, whose patience deserves a medal (or at least a very fancy dinner).

To my friends, who never let me forget that there is life outside academia and that laughter is indeed the best remedy for stress.

And a special shout-out to my carefully curated music playlist and endless cups of tea, for fuelling countless late-night writing sessions and keeping me warm and focused.

This one's for all of you – because without your love, support, and occasional comic relief, this thesis would still be a work in progress!

Contents

1. Introduction 1
2. State of the Art 3
    2.1 Mobility data collection 5
        2.1.1 Automatic Passenger Count – APC systems 9
    2.2 Transport data fusion 18
        2.2.1 Data fusion challenges 19
        2.2.2 Data fusion models 21
    2.3 Theoretical framework and literature gaps 29
3. Objective and methodology 37
    3.1 Definition of the study area 39
    3.2 Understanding available data sources 40
        3.2.1 Manual Passenger Count data. 40
        3.2.2 Automatic Passenger Count systems 40
        3.2.3 Environmental data 41
    3.3 Pilot study 45
        3.3.1 Definition of available data sources 46
        3.3.2 Design of low-cost APC system 47
        3.3.3 Data collection for pilot study 52
        3.3.4 Data processing for pilot study 55
        3.3.5 Data analysis and validation for pilot study 56
    3.4 Data collection and data pre-processing 57
        3.4.1 Manual Passenger Count data 57
        3.4.2 Camera-based APC system data 63
        3.4.3 Wi-Fi-based APC system data 65

3.4.4 Environmental data 65
3.5 Data processing 71
3.5.1 Importing datasets 71
3.5.2 Merging the CSV Files 72
3.5.3 Merging SHP files 73
3.5.4 Dataset segregation 73
3.5.5 Encoding and Normalisation 75
3.5.6 Resampling and Imputation 75
3.6 Data fusion models and validation 76
3.6.1 Validation metrics 77
3.6.2 Feature Selection 78
3.6.3 Statistical Models 79
3.6.4 Machine and Deep Learning Model 81
3.7 Comparative Analysis 83
3.7.1 Factors definition and calculation 84
3.7.2 Weights definition 85
4. Results 87
4.1 Pilot study 87
4.1.1 Model implementation in controlled environment 87
4.1.2 Algorithm refinement 89
4.1.3 Performance analysis 90
4.2 Data fusion 98
4.2.1 Feature Selection 98
4.2.2 Model tuning 103
4.2.3 Performance evaluation 104
4.3 Comparative Analysis 124
4.3.1 Factors evaluation 124
4.3.2 Weights calculation 129

4.3.3 Decision matrix 131
5. Discussion and Conclusions 133
6. References 139

List of Figures

Figure 1. Active infrared emitter on a bus 11

Figure 2. Passive infrared emitter on a bus 11

Figure 3. Treadle mat installed on a bus. 12

Figure 4. WIM system installed on a bus-stop. 13

Figure 5. Obstacle tracking method proposed by Li et al. using raw data fusion 22

Figure 6. (a) Original image sources vs. (b) Data fusion results 22

Figure 7. Methodology steps 38

Figure 8. GTT bus. 41

Figure 9. Sensors location for (a) meteorological data and (b) pollution data 44

Figure 10. Vector shape for Turin buildings 45

Figure 11. Pilot study methodology 46

Figure 12. Methodological framework for the low-cost APC system 47

Figure 13. System architecture 48

Figure 14. Low-cost APC system 49

Figure 15. Algorithm construction steps 50

Figure 16. Low-cost APC installation 54

Figure 17. Low-cost APC camera video frame 54

Figure 18. Output of the algorithm in the low-cost APC. 56

Figure 19. Sample Mobile Survey app 59

Figure 20. Sample of population information by gender in each census zone within the city f Turin as of 2023. 66

Figure 21. Territorial vector data for (a) census zones and (b) buildings 66

Figure 22. Pre-processing territorial variables 67

Figure 23. POI polygons and points 67

Figure 24. Pre-processing POI 69
Figure 25 Sample of data for Air humidity from the ARPA dataset 69
Figure 26. Pre-processing for meteorological data 70
Figure 27 Example of PM10 data from Torino - Grassi station 70
Figure 28. Pre-processing for pollution data 71
Figure 29. Processing methodology 71
Figure 30. Data fusion and validation methodology 77
Figure 31. Pre-Processing test 89
Figure 32 Object detection test 90
Figure 33 Object tracking and counting test 90
Figure 34. Highest overall accuracy: 25 October 2021 91
Figure 35. Lowest overall accuracy: 28 October 2021 91
Figure 36. Vehicle occupancy for the low-cost APC system over the whole 6-day period. 93
Figure 37. Vehicle occupancy (a) with original data and (b) with recalculated value 94
Figure 38. Vehicle occupancy for the three counting systems for the 2-day period 94
Figure 39. Vehicle occupancy for the three counting systems on (a) the uncrowded line and (b) the crowded line. 95
Figure 40 Feature selection - encoded but non-normalised DB - mutual info scores 99
Figure 41 Feature selection - encoded and normalised DB - mutual info scores 100
Figure 42 Feature Importance - Combined score - 4 methods 101
Figure 43. Scatterplot Simple Dataset 108
Figure 44. Best datafit graph - XGBoost 109
Figure 45. Worst graphs fit - XGBoost 110
Figure 46. Anomaly behaviour graphs – Simple dataset 111
Figure 47. Scatterplot Full imputation Dataset 114

Figure 48. Best datafit graph - LightGBM.....115

Figure 49. Worst graphs fit - LightGBM.....116

Figure 50. Anomaly behaviour graphs – Full imputation dataset.....117

Figure 51. Scatterplot Train imputation Dataset.....120

Figure 52. Best datafit graph - SARIMAX.....121

Figure 53. Worst datafit graph - SARIMAX.....122

Figure 54. Anomaly behaviour graphs – Train imputation dataset.....123

Figure 55. r2 score over sampling size for Statistical models: (a) Bayesian Regression and (b) SARIMAX.....126

Figure 56. r2 score over sampling size for ML models: (a) XGBoost and (b) LightGBM.....127

Figure 57. r2 score over sampling size for LSTM model.....128

Figure 58. Pair comparison matrix.....130

Figure 59. Final decision matrix.....132

List of Tables

Table 1. Review of commercial APC systems companies (Pronello, Baratti and Anbarasan, 2024) 15

Table 2. Data fusion models for passenger load estimation 31

Table 3. Meteorological data sensors - ARPA 42

Table 4. Pollution sensors location - aria ARPA 43

Table 5 Pilot study - Manual data collection structure - fields and description 55

Table 6. Manual passenger count's days collected 58

Table 7. Vehicles used for the study 59

Table 8. Data gathered with 1st survey - Front and Middle door 60

Table 9. Data gathered with 2nd survey - Back door 60

Table 10. ID variables for manual count records 61

Table 11. Column legend for final manual count file 62

Table 12. Column legend for GTT's APC data 64

Table 13. Column legend for Mobyforall APC data 65

Table 14. POI classes used for the study 68

Table 15. NaN values count, after merge, per time tolerance 72

Table 16. All features of main DB 73

Table 17. Controlled environment results 88

Table 18. Performance indicators for the controlled experiment 89

Table 19. Accuracy results from both systems 92

Table 20 Feature Importance scores - different methods and sum score 102

Table 21 Instances of each feature in the Top 20 ranking for different feature selection algorithms 102

Table 22 Hyperparmater tuning results for Prophet model - select few rows 103
Table 23 Hyperparameter tuning results for LSTM - select few rows 104
Table 24. Performance evaluation - Simple dataset 106
Table 25. Performance evaluation - Full imputation dataset 112
Table 26. Performance evaluation - Train imputation dataset 118
Table 27. Performance factors 124
Table 28 Scalability of models - some numbers 125
Table 29. Data quantity requirements 128
Table 30. Factors for Decision Matrix 129
Table 31. Letter assigned to factors 129
Table 32. Weight definition 130
Table 33. Inverted Normalization for factors 131
Table 34. Normalized decision matrix 132

Chapter 1

Introduction

Urban transport planning is a complex issue, and such complexity lies in several factors related to properly depicting the mobility patterns: the transport demand of passengers and freight has spatial (distribution of the trips) and time (period of the day) attributes but is also characterised by the purpose of travelling and by several psycho-social variables influencing the modal choice. Thus, understanding the mobility patterns requires, first, trip information and locations collected during different periods of the day. Therefore, to analyse the best combination of factors, good data collection and processing must be done.

Intelligent Transport Systems (ITS) contribute to a safer, more efficient, and environmentally friendly transport network. They make different aspects of urban mobility smarter, especially strategic planning, where they have made automated transport data possible. They can improve the operations of the entire transport network, leading to a better coordination of physical flows and resources (Mangiaracina *et al.*, 2017; Iliopoulou and Kepaptsoglou, 2019).

Historically, there have been multiple ways of collecting and understanding the mobility patterns of the users. Even though the most used tools have been the traditional methods such as travel surveys administered by telephone, nowadays the intelligent transport systems (ITS) implementation is bringing new technologies for both data collection and processing. Some examples of used technologies are mobile app applications that collect the data automatically, geographic information systems (GIS) for information about the territories, IoT devices, the inception of 5G mobile network connectivity, etc. In comparison with the traditional methods, these technologies reduce the costs of data collection improving the quality of data and increasing its quantity.

Therefore, tools like automatic passenger counting (APC) systems have been developed based on the latest technologies and have been offered to transport companies as a way to create indicators to measure their performance. However, there are no protocols or specifications that guide transport companies to choose the right system for them, arousing to sub-optimal systems application.

Since a large amount of data have to be collected from different sources, these datasets might be fragmented, and would not provide a full picture of the reality. Thus, new data analysis approaches have been developed to understand the key features of the data lake to properly use data in modelling the phenomenon. In this context, big data and learning models in a data fusion context are what matter. This research is aimed to understand how much a fused dataset can provide a better insight into the complexity of mobility patterns and to define guidelines to support transport companies in estimating on-board passenger count. This is achieved by collecting multiple sources data - including real operation bus data, meteorological and environmental data-, implementing multiple data fusion algorithms to estimate vehicle load and evaluating the results not only in terms of performance but considering also the dataset size, computational power and other factors that could affect the implementation of these models in real scenarios on the transport sector.

There is limited literature on the implementation and comparative analysis of different data fusion algorithms in the transport sector that uses manually collected operational data as ground truth. Hence, this study contributes to the field by collecting and analysing 57 days of real-operation data[1] and implementing various statistical, machine learning, and deep learning models to estimate passenger counts. Addressing the discrepancies between the claimed and actual performance of Automated Passenger Counting (APC) systems and exploring the modelling process as a continuous cycle, where each model is compared with real-world data to assess its applicability and representation of reality is challenging.

[1] The data were collected in a period on nine moths to cover different periods and seasons.

Chapter 2

State of the Art

The ITS Directive 210/40/EU defines Intelligent Transport Systems as “systems in which information and communication technologies are applied in the field of road transport, including infrastructure, vehicles and users, and in traffic management and mobility management, as well as for interfaces with other modes of transport”. ITS significantly contribute to a safer, more efficient, and environmental-friendly transport system. They can be applied in road, rail, maritime and air transport using different tools based on the complexity of the system. Among all the applications, the European Commission declared as primary priorities the traffic and travel information, the eCall emergency system, and intelligent truck parking within the union, across the borders between the member states, and in urban areas (European Commission, 2010)

ITS are the core of the Urban Mobility Package, developed by the European Commission. They can be widely adapted to different scenarios and used for different purposes and under different conditions, for both passengers and freights (European Commission, 2013). Moreover, it has been long recognized the value and potential of ITS data for strategic planning; using traditional methods, such as travel surveys, just allows for long-term transportation planning. In contrast, recent technologies developed for passenger count systems (APC), automatic fare collection (AFC) and automatic vehicle location (AVL) support real-time monitoring and data collection. That could enhance accurate and reliable transit demand modelling, mobility patterns and travel behaviour understanding (Iliopoulou and Kepaptsoglou, 2019).

Urban mobility can be described as a set of needs and constraints; needs of users and public authorities and constraints mostly due to funding and fragmented institutional background. However, environmental issues and technological advancement make urban mobility development very challenging. An increasing population result from the expansion of the cities; such as an urbanization process has resulted in an increase of the demand in urban areas (Fiorini and Lin, 2015). Additionally, it is necessary to promote a unified set of standards to ensure the compatibility of information technologies and information system in both transport and

infrastructure. Without established and consistent standards, any ITS is likely to experience fragmentation, inconsistency, and randomness.(Kabanov *et al.*, 2019).

Digital technologies are a significant driving force and enabler of various transport systems' improvements, they facilitate seamless data exchange among diverse stakeholders within the transport ecosystem, paving the way for real-time alignment of supply with demand. This not only optimizes resource utilization but also minimizes human error and fosters the integration of different transport modes into a unified mobility service, allowing multimodal transport systems. Furthermore, digital innovation in transportation is a boon for social innovation and broadens access to mobility, it introduces new market participants and innovative value creation models, such as the collaborative economy, underpinned by the evolution of vehicles into connected devices that interact with road infrastructure. For this reason, the European Commission has released a communication in 2016 explaining the strategy for deploying Cooperative Intelligent Transport Systems (C-ITS). This initiative envisions a future of cooperative, connected, and automated mobility, where digital technologies unlock new possibilities for information sharing. Such advancements promise significant improvements in road safety, traffic efficiency, and driving comfort by assisting drivers in navigating and responding to evolving traffic conditions effectively (European Commission, 2016).

On the other hand, with the use of diverse methodologies to visualise and analyse data, different insights from the ITS can be obtained because an integrated command and control systems can facilitate the use of dashboards to enhance the utilization of resources (Patel *et al.*, 2022). Hence, all the participants of the urban system are data generators: in data collection and processing the technology plays a key role.

Data collection technologies are evolving improving also the quality and reliability of the data. The generation of a huge amount of data with short updates leads to high interest to develop new strategies to manage, communicate, aggregate and analyse such data, also to support decision-making and control capabilities (Perallos *et al.*, 2015). Also, have contributed to developing new strategies regarding the privacy and data protection ; The development of new anonymisation techniques, and the importance of the application of high standards of cybersecurity in preventing cyber-attacks on the processed or stored databases will allow to manage and protect data. (European Parliament, 2018).

In 2021 the European commission launched Horizon Europe, a research and innovation funding program designed to contribute to the United Nations Sustainable Development Goals. The program seeks to enhance EU's competitiveness and growth by supporting the creation and dispersion of excellent knowledge and technologies. It also supports the implementation of EU policies while addressing global challenges. Among its five mission areas, one is dedicated to achieving climate neutrality, specifically focusing on "Climate-neutral & smart cities". (European Commission, 2021).

This particular mission emphasizes the importance of attaining climate neutrality by fostering the green and digital transformation of urban environments. Within this context,

transportation plays a significant factor due to its significant contribution to urban emissions, so is targeted for transformative changes such as the adoption of smart mobility solutions and the development of intermodal infrastructures (Boeri *et al.*, 2021; Cepeliauskaite *et al.*, 2021). Indeed, a transport system can be smart if it can deal efficiently with these new strategies. Thus, methodologies using data mining, machine learning and deep learning are emerging to achieve the desired smart transport systems.

In this literature review, the focus will be on ITS applications to urban mobility, regarding data collection and data fusion to identify the mobility patterns and estimate the demand in an urban context. First, a review of the data collection methods used in the transport field is presented and different automatic passenger counting systems (APC) are explained from both technical and commercial perspectives. Then, an overview of background knowledge of data fusion, its systems architecture and the challenges within the transport sector is provided. Next, a review of the recent literature about passenger load estimation in public transport systems and mobility patterns estimation in a real context is presented, analysing and discussing its performance to understand their efficiency in an urban environment.

2.1 Mobility data collection

The implementation of ITS brings with it new and discreet ways to collect the data through different hardware devices, such as mobile phone positioning (MPP), Bluetooth, global positioning system (GPS) and sensors. These methods allow easier and cheaper acquisition of long-term continuous trip data in comparison with the traditional methods whose higher constraints are related to high costs, infrequent updates, and small samples size due to the complexity of the collection itself. (Lee *et al.*, 2016). Some of the most used technologies nowadays for data collection in ITS and some fields for its application are described below.

MPP refers to the georeferenced data collected through the cellular networks; indeed, every mobile phone sends a signal to base transceiver stations (BTSs) that, after being processed, estimate the position of the user. Nevertheless, the level of precision will depend on the size of the cell. Uncertainty is higher in rural areas because the cells are larger; for this reason, this method is mostly used in urban areas. Despite this, issues like mode detection and misalignment of traffic analysis zones with BTS are faced (Lee *et al.*, 2016). For this reason, this type of data is usually processed with collections from other sources. As shown by Bachir et al. in (Bachir *et al.*, 2019), they inferred dynamic origin-destination (OD) flows by transport modes using mobile network data, transport network geospatial data, travel survey, census, and travel card data. On the other hand, this data typology does not require additional equipment but needs the cooperation of mobile phone companies.

Another type of data we can collect through mobile phones is their location data. Location data is more accurate than MPP and is collected more frequently due to its sampling rate which could be between 0.5 and 60 times per minute, allowing disaggregated trip data. The quality of route, travel time and OD data for use in travel modelling is higher in comparison with other

sources (Lee *et al.*, 2016). Nonetheless, the accuracy of satellite navigation systems (GPS, GALILEO, GLONASS, etc.) can be compromised in both urban and rural areas where signal interference caused by buildings, trees, or other obstacles can lead to gaps or incomplete trip tracking. This limitation is also noticeable in fleet management systems where poor satellite signals can impact the accuracy of estimated distance travelled data (Brlek *et al.*, 2022).

Location data can be collected for both passengers and freights, through users' mobile phone applications or directly from the public transport through its automatic vehicle location (AVL). In (He *et al.*, 2017), a system framework of traffic incident analysis and the prediction was designed using the GPS from mobile devices as a real trajectory dataset which was an input for their algorithm. Fang et al. (Fang *et al.*, 2016) proposed mining methods to discover people's mobility patterns from user GPS trajectories. In contrast, Zhu et al. (Zhu *et al.*, 2018) estimated travel time using AVL; they fused data from three different sources: bus-based GPS, inductive loop detector (ILD) and mobile phone network (MPN). In addition, mobile phone applications also provide the possibility of implementing arrival time prediction, bus crowdedness detection and automatic bus-stop detection on countries where ITS infrastructure are limited (Rajput *et al.*, 2022).

Vehicles equipped with dynamic route guidance (DRG) can generate a type of data called "floating car data". DRG acts as roving traffic detectors, just like the vehicles from the freeway service patrol or road watch which are specialized and equipped to perform this task and can be transmitted to a traffic centre through an onboard communication device. These vehicles, which are capable to determine experienced traffic conditions and communicate them, and are active participants in the traffic flow, are called probe vehicles (PVs) (Westerman, 1995). From probe vehicles, travel time data can be extracted to improve traffic signal programs(Fiorini and Lin, 2015). For instance, Chang et al. estimate reliable traffic speed to improve the decision-making in emergency situations (Chang *et al.*, 2014) using space speed data collected by PVs, stationary vehicle detectors (VDs) and historical data.

As a result of the endeavours to advance C-ITS, automated and connected vehicles are actively involved in generating additional data, revealing new possibilities. This data collection supports the ambition to decarbonize the transport system by aligning with C-ITS objectives such as enhancing traffic efficiency, optimizing urban infrastructure use, and minimizing fuel consumption and the environmental impact of transportation. (European Parliament, 2018). For instance, strategies have been proposed for a driver-vehicle-infrastructure integration for real-time data acquisition to aid eco-driving (Vojvodić *et al.*, 2017), alongside advancements in the development of an Advanced Driver-Assistance System (ADAS) for more accurate and reliable on-vehicle sensing using real-time Radar and Lidar sensors (Giuffrida *et al.*, 2023). Other research focuses on reducing fuel consumption and emissions in scenarios with stop-and-go traffic by determining an optimal vehicle speed to avoid unnecessary idling at traffic lights (Pariota *et al.*, 2019; Sharara *et al.*, 2019; Hepner *et al.*, 2021) Furthermore, efforts to refine electric vehicle aggregators' (EVA) ability to predict their base power rate and the flexibility in demand response have been undertaken. These enhancements aim to exploit opportunities

in energy arbitrage and ancillary services, offering significant benefits in terms of energy efficiency and grid reliability for electric vehicles. (F. Lu *et al.*, 2021; Gong *et al.*, 2022)

Closed-circuit television (CCTV) has long been a staple in data collection and, thanks to advances in processing methods, continues to be relevant. These cameras are often deployed across highway networks, particularly at vital junctions, to monitor traffic. An example of such application is the Transport Operation and Information System (TOPIS) in Seoul, South Korea, this system can monitor real-time traffic conditions and can detect traffic congestion by using more than 700 surveillance cameras installed throughout the city. (Macheret *et al.*, 2021). Although leveraging CCTV cameras alone for urban traffic management can be challenging, their utility significantly increases when combined with other data sources; for instance, integrating camera data with that from multiple sensors to estimate Origin-Destination (OD) matrices has proven effective. This method of data fusion enhances traffic control and planning (Biletska *et al.*, 2009).

Nowadays to improve driving experience, modern vehicles have on-board cameras monitoring the surrounding environment also in view of the development of automated vehicles (AVs). The data coming from these on-board cameras and CCTV have been collected and used for vehicle recognition to estimate the number of vehicles, their average speeds and to track and classify them (Kavitha and Chandrappa, 2017). Deep learning techniques have been used for the above activities and image recognition algorithms such as YOLO, CNN, KNN and Haarcascade have been improved and trained for traffic estimation (Chen *et al.*, 2021; M. Lu *et al.*, 2021; Pavani and Sriramya, 2022). Furthermore, different evaluation metrics have been proposed to promote the appropriate advancement and rigorous quantitative evaluation of the reliability and performance efficiency of recognition models when applied to video data (Panetta *et al.*, 2021).

Sensors play a crucial role in various ITS applications. In the context of monitoring urban roads, a laser-based traffic detector has been developed to estimate key parameters, such as speed, length, and height of vehicles, to prevent potential collisions. (Qiu *et al.*, 2021). Authors have proposed diverse methods to perform passenger counting in lanes devoted to High-Occupancy Vehicles (HOV), using near-infrared cameras with specific spectral sensitivity and thermal images (Pavlidis *et al.*, 1999; Nowruzi *et al.*, 2019). Moreover, the city of Hong Kong use sensor wires embedded beneath the asphalt, as well as video surveillance systems, to accurately determine traffic conditions. These data are then utilized together with a unified light control system to optimize traffic flow and enhance overall road efficiency (Macheret *et al.*, 2021). On the other hand, sensors can be used to ensure safety in the transport of goods as well; for example, 8 different sensors (gas, temperature, humidity, pressure, liquid level, acceleration, angle and switch sensor) were implemented in a tank for liquefied natural gas (LNG) (Chen *et al.*, 2012) to collect data in order to automatically predict the dangerous states of hazardous chemicals' transport.

Merely collecting location and sensor data falls short in terms of characterizing users according to their specific characteristics; to this end, additional methods such as travel surveys, smart cards, or automatic fare collection (AFC) systems are required.

A travel survey is designed to collect information about the characteristics of the travel – when, where, how, the reason why – and to monitor long-term trends in travel behaviour. Surveys collect also the factors that affect travel, like as the user is a driving license holder or his/her profession and status. The traditional way of collection provides that users fill in a travel diary, on paper format, or answer phone calls or interviews to make known his/her travel activities during a day or for longer periods. Because this information relies on the memory and judgment of participants and due to its high implementation costs, this method is no longer able to address the new modelling needs (Safi *et al.*, 2017). For this reason, new surveying techniques have been developed using the technologies as a mediator:

- Web-based data collection method, a diary-based survey in which participants fill a web-based questionnaire to report their travel activities.
- H-tracker method, it is a combination between a GPS device and a web-based questionnaire. The users should keep with themselves a GPS device and at the end of the day they label the collected data using a web-based questionnaire, with information such as mode, purpose, departure, and arrival;
- S-tracker method, it is similar to the H-tracker method but, instead of using two platforms, it performs both data collection and labelling through a smartphone application. The users should install this application on their phones. Then the data is sent to a server where it is processed;
- ATLAS II method, it is the evolution of the S-tracker. Stands for Advanced Travel Logging Application for Smartphones, it is a mobile application and, to be used, the users should install it on their phones and, at the end of the day, they have to label their activities. In comparison with S-tracker, ATLAS II is more accurate and has real-time data processing to clean collected data. Also, it has trip visualization and a labelling reminder, which allows a better recall of users' activities during the day.

It is worth mentioning that researchers have asserted that ATLAS II exhibits superior data accuracy and improved data quality compared to other methods, attributing this to the system's trip visualization and reminder features that allow users to label their trips more precisely. These techniques were also utilized in the New Zealand National Household Travel Survey to facilitate demographic attributes analysis. Here, the authors noticed that younger participants were more interested in the smartphone-assisted methods, while the older participants were more comfortable with web-based methods or H-tracker. Similarly, the United Kingdom is looking to update its national travel survey (NTS) (Safi *et al.*, 2017). An extensive evaluation of both current and emerging technologies and data sources to improve its NTS, suggested that a diversified approach, including web, computer, and smartphone tools, would most effectively capture a representative sample of the population. This strategy recognizes the varying preferences across age groups, ensuring inclusivity of both young and elderly individuals in the survey. (Transport System Catapult, 2017)

Regarding the AFC data, it is usually collected from smart cards and tickets with RFID tags by validations machines inside the bus. Once a transaction is generated, the time and location are recorded for further studies in transport and urban planning. The use of it can offers

insights on urban dynamics and mobility patterns of public transport users as it provides individual users' demographic data such as age, gender and income. For instance, AFC data has been used for individual trip reconstruction for OD matrices estimation and agent-based transport models and for modelling the passenger arrival process to transit stops (Anda *et al.*, 2017; Kieu and Cai, 2018; Hadas *et al.*, 2022).

Despite in some literature just one data source is used for O/D matrix estimation, the fusion of different sources can lead to an improved model. It means that the issues from a specific data source can be overcome by another data source, i.e. if we fusion location data and smart card data, we can estimate a location with higher accuracy. There could be problems related to lost or weak satellite signal that can be solved with the smart card data location, and the same way around. Multi-sensor data fusion compensates the shortcomings of individual sensor sources, increasing robustness, confidence, and spatial coverage of the input estimation, increases the confidence and reduce the ambiguity (Han, 2012).

Although the hardware component is a critical part of traffic management and provides the foundational data for intelligent transport system functions, dynamic traffic management systems will experience a fastest growth driven by the models and real-time traffic information, allowing to share information, predict the impact of different traffic management strategies, and, definitely, to have a more efficient traffic management. Despite hardware sales dominated the traffic management market in 2019, the focus is expected to shift towards dynamic traffic management systems in the near future (Macheret *et al.,* 2021).

2.1.1 Automatic Passenger Count – APC systems

Automatic passenger count systems were invented to support on-boarding records of frequently used public transports and, boarding and alighting data that helps to records all the information about their passengers. Because of its accuracy and reliability, it is commonly used in transit systems and buses for the count of flow of passengers and mobility patterns estimation. As this information is relevant for transport management to distribute their assets, decrease the cost of transport and increase service quality, public transport companies have a big interest in counting passenger. In addition, federal agencies require ridership reports due to government legislation. Thus, the implementation of APC systems may offer benefits to both transport systems and federal/state funding agencies (USA department of transportation, 2021).

APC systems are strongly related to crowdsensing technologies. In both, people are considered and transformed on active actors. Some studies have tested methods that invite people to wear specific devices to collect data from them. However, the acceptance is low due to the invasive nature of such devices and the costs to develop them (Fernandes *et al.*, 2018). Apart from the use in the transport field, these types of counting tools are also used in the retailer and buyers' analysis. Retail stores can learn about the popularity of the products in different aisles if they know buyers' speed/density in different parts of the store (Nogueira *et al.*, 2019). However, the literature is going to focus on the applications of APC systems in the transport sector.

In the transport sector, these systems provide diversified information that, from the nature of the system, is easily transmitted. This huge amount of data is produced in a continuous flow of information. Moreover, the collected information could be correlated to get an overall view of a specific phenomenon and could be paired up with the automatic vehicle location systems (AVL) to give better insights on the load status, bus real timetables and a continuous monitoring of the lines. Transport companies seek to increase efficiency by collecting information from a variety of sources (Butler *et al.*, 2020; Dimitrakopoulos *et al.*, 2020). A number of companies have chosen to install APC systems in their vehicles in order to better estimate vehicle occupancy and to have more accurate information regarding the boarding and alighting of users.

To make effective decisions in service provision that will increase efficiency and improve the provision of transit service, it is critical to collect and evaluate passenger boarding and alighting count. So, the first step in implementing an APC system in a transport system is to understand and evaluate the intended use of the data, which could be used for service planning, scheduling or transportation management (Baltes and Rey, 1999):

- Data for service planning: Related to daily variation in total passengers, county operation, overcrowding times/locations, boarding and alighting, and real run times in comparison to scheduled ones. With this data, minimum and maximum passenger loads by direction can be monitored and average speeds can be determined. Useful data to infer possible seasonal variations in ridership and run times. The main requirement is to have a significant amount of data to ensure statistical significance.
- Data for scheduling: data related to individual stop, run and dwell times, trip boarding and alighting, and on-time performance. The idea is to monitor and in case adjust trip schedules to better match supply and demand to maximize both cost efficiencies and passenger comfort.

Different commercial systems using different technologies are available: (i) infrared technologies, (ii)floor-based sensing, (iii) optical solutions and (iv) mobile-based sensing solutions.

i. Infrared sensors

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. It can be divided into two categories: Sensors of the active type and sensors of the passive type. In the active infrared sensing process, laser diodes emit low-power infrared energy in the near-infrared range of the electromagnetic spectrum to light up the detection areas. While in the passive sensing process, sensors are capable to detect infrared radiation, but they do not emit it using a LED.

There are different active infrared sensing devices, however the photoelectric detection technology is more stable among them and for this reason is widely used (Song *et al.*, 2008). These can be found on the infrared beams counting system; Infrared beams count the

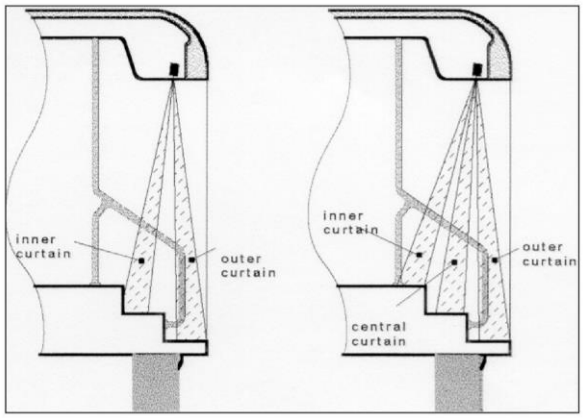
passengers when they pass between the beams, they are mounted either horizontally or vertically and directed across the path of entering and exiting passengers (Figure 1).



Source: (Pinna and Dalla Chiara, 2010)

Figure 1. Active infrared emitter on a bus

Literature shows how active infrared sensors are used for different counting mechanisms, not only in the transport field. For instance, authors have proposed a customer-counting system based on RBF neural network using technology of infrared photoelectric sensor (Song *et al.*, 2008). Passive infrared sensors usually use pyroelectric technology which consists of a method of detecting changes in temperature. This means, that the sensors will detect the infrared radiation emitted by a human body. There are APC systems developed using this technology; for instance, the initial INIT Infra-Red Motion Analyzer (IRMA) is a system that utilizes passive infrared sensors with integrated optics to detect and record the thermal image of individuals as they pass through a door; it is mounted overhead in the door frame (Figure 2) (Gerland and Sutter, 2013).



Source: (Gerland and Sutter, 2013)

Figure 2. Passive infrared emitter on a bus

However, nowadays these systems are usually coupled with another technology. INIT, a well-known innovative transport company, has a catalogue of APC products which use infrared sensors with video-based systems, or both active and passive sensor in the same device

(Rakebrandt, 2013). Similarly, in literature, authors have proposed a low-cost APC that combines the use of infrared sensors, a camera and computer vision algorithms (Vieira and Júnior, 2022).

In general, infrared-based APC systems are known for their cost-effectiveness, non-intrusiveness, ease of installation and ability to provide real-time and highly accurate passenger data. Despite being highly accurate, the effectiveness of these systems may be affected by some constraints such as unfavourable lighting conditions, environmental variables like high temperatures, dust, and humidity, and the possibility of failing to detect clusters of passengers when they overlap while entering the vehicle.

ii. Floor-based sensing

Floor-based sensing use sensors installed on the floor to detect the weight and movement of passengers or vehicles. The main two technologies used in the transport sector is Treadle mat sensors and Weight In Motion (WIM) systems.

The treadle mat sensors are devices that can estimate the passengers boarding or alighting by detecting the pressure on a mat; The mat is usually installed at the entrance of the bus (Figure 3), and it is essential to evaluate and experiment with its placement thoroughly to avoid any inaccuracies in detecting data. During the evaluation of this system in actual applications, authors have faced challenges when multiple passengers pass the sensor simultaneously, as well as when passengers stay stationary on the mat for a prolonged period. They have also verified that environmental variables, such as vibrations and dust, can impact the system's accuracy, and there is a potential danger of harm if fluids penetrate the mat (Zhu *et al.*, 2008; Butler *et al.*, 2020; Mccarthy *et al.*, 2021)



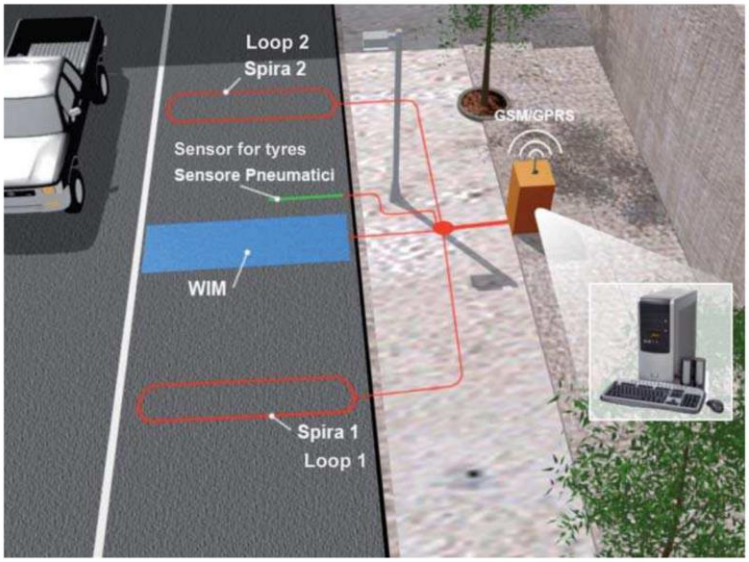
Source: (Tapeswitch, 2020)

Figure 3. Treadle mat installed on a bus.

WIM systems can estimate both the total weight of a vehicle and the weight distribution across each wheel assembly. This is accomplished by using multiple sensors placed in the road

surface which capture the dynamic forces exerted by the vehicle's wheels as it passes over them. It is a method that has been used for monitoring large infrastructures but is under study for developing “smart roads”. Its accuracy depends on the vehicle dynamics, pavement integrity and composition and system calibration (USA Federal Highway Administration, 2014; Chen *et al.*, 2018; Gallina *et al.*, 2022).

In the context of APC applications, WIM systems can estimate the total number of passengers on-board a vehicle, but they cannot count individual passengers boarding or alighting the vehicle. They are often used in conjunction with other technologies, such as proximity switches and communication systems like modems, GSM, or GPRS. To get a complete count of passengers, WIM systems are typically installed at bus stops (Figure 4), and the data they collect is transmitted to a central system via the communication system. This data can be combined with information from other sensors or counting methods, such as infrared or video-based systems, to provide a more accurate count of passengers on a particular bus or route. The resulting data can be used for various purposes, such as scheduling and resource allocation, fare collection, and demand forecasting.



Source: (Pinna and Dalla Chiara, 2010)

Figure 4. WIM system installed on a bus-stop.

iii. Mobile sensing-based solutions

Smart mobile devices with wireless networking are available everywhere and usually have features that can be used to estimate the number of passengers on a bus, i.e. Wi-Fi, Bluetooth and internal sensors. Once these devices have any of these features on, emits a signal that can be ‘listen’ by a receiver.

The APC system consists of one or more terminals installed in the vehicle that are capable of detecting and counting both Wi-Fi and Bluetooth signals. When a passenger carrying a smart device enters or exits the vehicle, their signal is sensed and recorded, and eventually the number of passengers can be estimated according to the number of active devices. Some solutions collect probe request frame, or measures the signal strength, from on-board devices and matches their MAC addresses in order to increase or decrease the count. (Myrvoll *et al.*, 2017; Mehmood *et al.*, 2019; Mccarthy *et al.*, 2021). However, as companies are enhancing privacy and security on their products by implementing randomized MAC addresses, new challenges arises on de-randomizing techniques (Grgurević *et al.*, 2022). Solutions like iABACUS have been developed to try to overcome this and allow a higher counting accuracy (Nitti *et al.*, 2020).

Another APC mobile sensing-based solution consists on a mobile application that collects information directly from the cell phone, mainly for crowdedness detection. On these solutions, both GPS and accelerometer sensors are used, and they can reach an accuracy between 80% and 90% percent for bus-crowdedness detection (Rajput and Chaturvedi, 2019; Rajput *et al.*, 2020).

iv. Optical solutions

APC optical solutions include the video counting systems and stereoscopic cameras implementation. These technologies measure passenger volumes and boarding and alighting of passengers. In most cases, a number of different algorithms are used for detecting motion, estimating the direction of the identified object, and confirming the existence of a moving passenger (Olivo *et al.*, 2020). These APC systems are usually mounted above the doors to count passengers entering and leaving the vehicle; however, alternative solutions for counting passengers have been developed by employing Closed-Circuit Television (CCTV) cameras placed within the vehicles. The prevalent characteristic in both approaches is the zenith positioning of the cameras, i.e., above passengers' heads (Kuchár *et al.*, 2023).

In Europe, seven major companies produce and sell APC systems, either as manufacturers or system integrators. The main products available on the market are shown in Table 1. Manufacturers produce and sell hardware components for the APC systems, while system integrators buy APC devices and incorporate them into commercial packages, offering services such as video surveillance and smart ticketing in addition to the devices themselves.

Some of these companies meet the VDV 457-2 accuracy requirements (IRIS intelligent sensing, 2016; HELLA Aglaia, 2022). VDV stands for Verband Deutscher Verkehrsunternehmen, the association of German transport companies that publishes recommendations and general advice concerning the implementation of APC systems in different vehicles so as to ensure the quality and uniformity of the data collected and consequently the reliability of projections based on these data. VDV 457-2 sets out rules for verifying and certifying counting accuracy (VDV Industrieforum, 2018). Nonetheless, no comprehensive universal standard for measuring passengers onboard in public transport has so far been established. As an illustration of this absence of a common standard, countries including the United Kingdom, the United States, and Australia employ a variety of different

methodologies to assess rail crowding. Some rely on the load factor, while others estimate the rolling hour average loads, or count passengers in excess of capacity (Hensher, 2020).

As can be seen in Table 1, the accuracy and precision of the different optical-based solutions listed is claimed to be between 98 and 99 percent in every case. However, these indicators are deceptive, since they are usually obtained under ideal conditions, often in a depot or in a laboratory test. In addition, the accuracy of an estimation is sensitive to camera angle, passenger-flow density, and lighting conditions. If any of these is subject to variation, accuracy will rapidly decline (S. Wang *et al.*, 2021).

Table 1. Review of commercial APC systems companies (Pronello, Baratti and Anbarasan, 2024)

COMPANY	TECHNOLOGY	DEVICE	ACCURACY – PRECISION	BUSINESS MODEL
EUROTECH	Stereoscopic vision with IR sensors	DynaPCN 10-01 DynaPCN 10-20-00 DynaPCN 10-20-01	98% - 99%	Manufacturer
SELSAT (Selecta Digital Service)	Stereoscopic vision with IR sensors	DynaPCN 10-01	98% - 99%	System Integrator
AESYS	Stereoscopic vision	Aesys VideoSIGHT	98%	System Integrator
TRIPPEOPLECOUNTER	WiFi solution	-	Percentage error: 10%	System Integrator
HELLA AGLAIA	Stereoscopic vision with IR sensors	APS-R APS-B	98%-99%	Manufacturer
DILAX INTELCOM	Triangulation method Stereoscopic vision	DILAX IRS-320R DILAX IRS-400	99%	Manufacturer
IRIS Intelligent Sensing	3D IR matrix sensor (ToF)	IRMA Matrix	99%	Manufacturer

Researchers have been developing solutions to improve APC's estimations and passenger count systems based on image and video processing. Nasir et al. (Nasir *et al.*, 2018) developed a methodology for estimating the number of passengers on a bus, which involves processing on-board bus images using several image processing techniques including colour conversion, image segmentation, and the removal of noise. Likewise, some authors (Liciotti *et al.*, 2017; Che Husin and Abdul Nasir, 2021; Kuchár *et al.*, 2021) have used image processing for

estimating the passenger count by attempting image segmentation on different colour conversions or with the help of depth/thermal cameras.

Other passenger counting techniques have been proposed that harness ML and DL algorithms. For instance, Adaboost classification of histograms of oriented gradients and colour histograms has been used to detect people and a Kalman filter to track them (De Potter *et al.*, 2012). Furthermore, neural networks (NN) in passenger counting systems have shown significant promise. In the study of Liu et al. (Liu *et al.*, 2017), a convolutional neural network (CNN) and a spatio-temporal context model were used to detect passengers and to track their moving heads. Kumar Singh et al. (Kumar *et al.*, 2021) proposed a single shot detector (SSD) mobile net along with a centroid tracker for the improved extraction of features and classifications, while Hsu et al. (Hsu *et al.*, 2020a) combined two deep learning methods to estimate passenger occupancy in different scenarios: a convolutional autoencoder to extract features from crowds and determine the number of people, and the 'You Only Look Once' version 3 architecture (YOLOv3) for detecting the area of a bus in which head features are clearest. Similarly, Seidel et al. (Seidel *et al.*, 2022) proposed an end-to-end Long Short-Term Memory network for automated passenger counting using a privacy-friendly dataset, only containing depth information from low-resolution 3D LiDAR video recordings; the study claimed an average accuracy of 96% for both boarding and alighting.

A number of applications in the literature combine an object detection algorithm with a tracking algorithm for counting and identifying when a passenger is boarding or alighting. Valencia et al. (Valencia *et al.*, 2022) explored the use of different YOLO algorithms combined with a DeepSORT tracking algorithm for overhead people detection and count; their findings indicated that Tiny-YOLOv4 provides better accuracy with only few false predictions. Meanwhile, Zhang et al. (Y. Zhang *et al.*, 2020) combined two-class SSD model for passenger detection, with a Kalman filter for tracking purposes. They found that DL algorithms yield significantly superior accuracy in comparison to traditional image algorithm methods. Likewise, by combining YOLOv2 and an MIL tracker, Liu et al. (L. Liu *et al.*, 2020) proposed an algorithm for counting the number of passengers boarding and alighting in real time, claiming an accuracy of up to 99%. To improve the object detection performance, Zhang et al. (S. Zhang *et al.*, 2020) proposed a modification on the convolutional layer of a tiny YOLO network to effectively improve the operational speed without compromising the detection accuracy.

Other NN algorithms used in the literature are multi-column CNN for estimating a density map of the passengers' heads and counting them (Liang, Qian and Zhu, 2021; Moreno Rendon *et al.*, 2023). On the other hand, some authors have combined both ML and DL methods. Nakashima et al. (Nakashima, Arai and Fujikawa, 2019) used a YOLOv3 + DeepSORT algorithms to estimate the number of passengers boarding and alighting from a CCTV camera installed on a bus, and then proposed a correction method by applying a Random Forest model. This method improved the estimation by 2.7%: the average accuracy of the image processing estimation was 93.5%, and this rose to 96.2% after application of the correction method. Similarly, other authors have focused on low-cost systems that utilize existing surveillance

cameras installed on buses and process videos off-line. These authors have tested different ML and DL algorithms for passenger detection (Grönman *et al.*, 2019; Hsu *et al.*, 2020a, 2020b; Guo and Zhang, 2022). In addition, there have been attempts to enhance existing APC estimations: some authors have tested different algorithms for real-time passenger counting (Jenelius, 2020a, 2020b; Amato *et al.*, 2022; Hangli *et al.*, 2022; Hoppe *et al.*, 2023), while others have introduced methods that can be applied in a post-processing phase (Hu *et al.*, 2020; Roncoli *et al.*, 2023).

In the literature, outside the field of transport, there are studies focusing on people detection and counting that employ similar approaches based on overhead video cameras. Some authors have used Kalman filtering for tracking purposes (Hongmei *et al.*, 2020) and to improve a YOLO-based object detection and tracking algorithm (Vignarca *et al.*, 2021). Others have used only object detection algorithms for counting (Ahmad *et al.*, 2019a, 2019b).

Alongside the development of new passenger detection algorithms, there has been some work in developing low-cost APC devices for public transport applications. The cost of a commercial APC is between EUR 1500 and EUR 3000 per door, and generally it is supplied as part of a service with a monthly fee (Information obtained from verbal exchanges with several transport companies). Some devices from the literature have a cost that is half that of commercial products, while providing an equivalent or superior performance in in-field applications. In Sydney (Australia) the metropolitan public bus services conducted an experiment over a period of seven days, including weekdays and weekends, to evaluate different APC solutions. One of these was a real-time-performance video-based passenger counting system (Mccarthy *et al.*, 2021) with a total cost of approximately AUD 560. This system featured a Raspberry Pi 3B+ as a microprocessor with a Neural Compute Stick connected, and a Raspberry Pi Module V2 camera. Regarding the software, for the detection algorithm, the authors used only MobileNet SSD, a CNN, and the Intersection over Union (IoU) calculated between two consecutive detection bounding boxes as the tracking algorithm. Their system obtained an overall accuracy of 57%. Kniess et al. (Kniess *et al.*, 2021) developed a counting scheme with similar hardware, but they added an aggregation algorithm for managing data-transmission strategies. The detection and tracking algorithm were executed on-board while the classifier ran on an external computer. This system was less successful than YOLOv3 in identifying boarding and alighting by users, although YOLOv3 has a long count processing time (800–1000 s). In contrast, Recalde et al. (Recalde *et al.*, 2021) developed a counting system with a boarding and alighting accuracy of, respectively, 89% and 88%. A comparison of vehicle load estimates from both commercial and low-cost APC systems suggests that current APC systems may not meet expectations under real-world conditions, whereas low-cost systems might achieve comparable or even superior accuracy compared to very expensive commercial systems (Pronello and Garzón Ruiz, 2023).

The surge in new technologies like artificial intelligence (AI) and deep learning (DL) algorithms in recent years has significantly increased the reliance on camera-based systems across various sectors, including transportation, retail, manufacturing, and healthcare. This

trend reflects the growing applicability and effectiveness of these technologies in diverse fields. (Paneru and Jeelani, 2021; Dhou *et al.*, 2022; Munteanu *et al.*, 2022; Oroceo *et al.*, 2022)

2.2 Transport data fusion

Data fusion (DF) is a technology that allows getting more informative, consistent, and accurate information from imprecise, inconsistent, and conflicting data by merging raw datasets. It is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making and involves integrating data from multiple sources to create a comprehensive and accurate representation of a state or situation (Boström *et al.*, 2007; Jain *et al.*, 2021). Thereby it has so many applications in different fields.

In the literature, both fusion and aggregation methods are discussed; it is crucial not to confuse them as they serve different functions. Data aggregation involves combining data from various sources into a cohesive summary view, typically by grouping data based on common attributes such as time, location, or category. Common aggregation methods include arithmetic functions like sum, average, minimum, and maximum. In Intelligent Transportation Systems (ITS), a range of sensors collects data, employing diverse communication technologies such as Wireless Sensor Networks (WSNs). Data aggregation algorithms in these contexts primarily aim to eliminate duplicate data collected by multiple sensors and to minimize energy consumption when transmitting results to a central processing unit, known as the sink node. These algorithms are also utilized during routing to optimize the path to the sink node, thereby enhancing resource sharing (Eskandari and Ayughi, 2011).

Conversely, data fusion involves constructing a system in which different components interact and integrate. This system typically includes data collectors, databases, data fusion algorithms, and visualization platforms, all of which continuously communicate and exchange information. It is a complex process that automates the detection, association, correlation, estimation, and integration of data and information from single and multiple sources to refine estimates of position and identity, and to provide comprehensive and timely assessments of situations along with their impacts and significance. (Bass *et al.*, 1997; Klein, 2022).

Due to data incompleteness, conflict, and inconsistency raw data collections usually are not suitable as input for data fusion algorithms, they do not allow extracting information with high value at once. Consequently, a hierarchical transformation is needed to deal with data imperfection and to manipulate data systematically. Since data fusion is a complex system constituting many parts to process data, a widely spread data fusion architectures novels have been introduced to unify expressions and terminologies for describing each part's functionalities and characters. Including Joint Directors of Laboratories (JDL) (Llinas *et al.*, 2004), the Luo and Kay architecture (Luo *et al.*, 2002) and the Dasarathy's architecture (Dasarathy, 1997).

Traditional data fusion techniques include evidential belief reasoning fusion, probabilistic fusion, and rough set-based fusion (Castanedo, 2013). For instance, a data fusion method for the IoT that integrates Dempster-Shafer theory with an adaptive weighted fusion algorithm (Boulkaboul and Djenouri, 2020). However, the development of sensors, processing hardware, and new technologies, are bringing the opportunity to develop innovative data fusion techniques by using machine learning and deep learning models. Moreover, attribute and identity estimation form the fundamental basis of the data fusion field, and are primarily influenced, respectively, by distinct advancements in statistical estimation theory and machine learning (Schmitt and Zhu, 2016).

Machine learning is a form of artificial intelligence that enables a system to learn from data in order to gain a new level of understanding. In an iterative process, the algorithm can learn from data to improve, describe, and predict outcomes. As a technique with strong abilities to compute and classify data, machine learning is highly expected to improve the overall performance of data fusion algorithms (Meng *et al.*, 2020). On the other hand, deep learning is an extension of machine learning; it uses multiple processing layers with complex structures to model high-level abstractions in data. It is very common to find DL applications in autonomous driving (Pouyanfar *et al.*, 2018; Thikshaja and Paul, 2019).

As a result of the diversity of applications and architectures, data fusion algorithms can be grouped into three levels based on their final use: (i) Signal level data fusion which is the lowest level of data fusion and is generally applied in signal fusion, image fusion (or pixel fusion) and raw data related fusion applications; (ii) feature level data fusion whose output can obtain characteristics or features in the form of patterns; and (iii) decision level data fusion, that is the highest level, where is involved the fusion right before deciding on a task. (Meng *et al.*, 2020). Section 2.2.2 presents and classifies on these levels different data fusion applications in the transport field.

2.2.1 Data fusion challenges

Transportation systems look forward to maximizing the efficiency and safety of the roadway network thus requires accurate and reliable data for monitoring and managing operations. Similarly, the flourishing of automated and connected vehicles has created new scenarios to collect a high amount of data regarding vehicle volumes, congestions, collision avoidance and potentially dangerous road conditions. Therefore, a major opportunity and challenge is identified to support the implementation of ITS in both public and private sectors.

Opportunity on creating big collections of relevant and diverse data for traffic and transportation management. The challenge arises from the need to amass, process, analyse, and store an extensive amount of data that will flow and be used by traffic management agencies. Besides, using technologies like Bluetooth, mobile phones and Wi-Fi, transport users are also part of this big data ecosystem.

As the data ecosystem continues to grow, it must be properly handled to give useful insights into the intricacies of traffic operation. As a consequence, the challenge of developing data fusion methods, that can cope with these data, emerges (Klein, 2019).

Due to its complexity, multiple issues make data fusion a challenging task. The implementation of ITS applications and the concurrent need for real-time and accurate data supporting traffic management functions such as route guidance, incident detection, demand management, and safety warning on automated vehicles, shows the importance of complemental data for traffic flow parameter estimation (Faouzi and Klein, 2016; J. Liu *et al.*, 2020; Cecaj *et al.*, 2021). In the transport sector, every single application faces different needs based on the number of data sources, data heterogeneity, and the results response time. The main data fusion challenges within the transport sector are the following:

- *Data sparsity and imperfection:* The data collected is often vague, ambiguous, and incomplete, particularly the data captured by sensors, which in an urban context are frequently imprecise and uncertain. This can lead to reduced performance and an increase in estimation errors. However, by utilizing complementary information and mathematical tools, the quality of these datasets can be improved, although some errors remain challenging to estimate. Both data sparsity and imperfection are major issues that all data fusion methods are expected to address. These challenges can seriously affect fusion quality, as useful and important information may not be extracted if the data fusion algorithm cannot effectively exploit data redundancy to mitigate their effects.
- *Data scarcity:* In the context of relatively new urban areas or cities that lack substantial historical data, constructing accurate models or utilizing extensive historical datasets for learning becomes challenging. However, in such situations, it is possible to compensate for the lack of local data by drawing on knowledge and data from similar circumstances in other cities. This approach leverages the insights and patterns observed in comparable urban environments to overcome the limitations posed by data scarcity. Nevertheless, there is still a level of error as transfer leaning processes are not easy generalizable and therefore present a challenge that requires further study.
- *Data inconsistency:* Outliers and spurious data can be generated by inherent noises in measurements, sensors, and external factors. In addition, lasting and dynamic failures may happen due to the complexity of the system. Therefore, DF technique should overcome this problem by eliminating the influence of data inconsistency
- *Data alignment/registration and multi-modal data:* Data is collected from various sources and environments, each with its own typology suited for different purposes. For instance, spatio-temporal data differs from smart card data in both structure and usage and multi-modal data may be available in an urban context. Similarly, data collected from different sources has its own local frame, such as time zone and spatial properties. Before using these datasets in the model, they should be converted to a pre-defined format to ensure that DF methods can accurately describe the whole status of

a system. Fusing different types of data with high-dimensional features and selecting the most effective features for the current task is a problem that requires further study.

- *Fusion location:* Even though distributed fusion environments can fuse its datasets in a central or local node, aspects like costs, bandwidth, answer time and information loss must be considered during implementing a DF model. Finding the balance fusion cost and fusion quality is a challenging matter.
- *Dynamic fusion:* To estimate a time-varying system state, data might be significant for only limited time. The complexity of DF comes by using the same time-frame period for all the datasets, especially for real-time application environment. On the transport sector, many of the DF models are implemented for getting a rapid result. Thus, to guarantee good quality results, the fusion node should be able to perceive the right order of data and its validation.

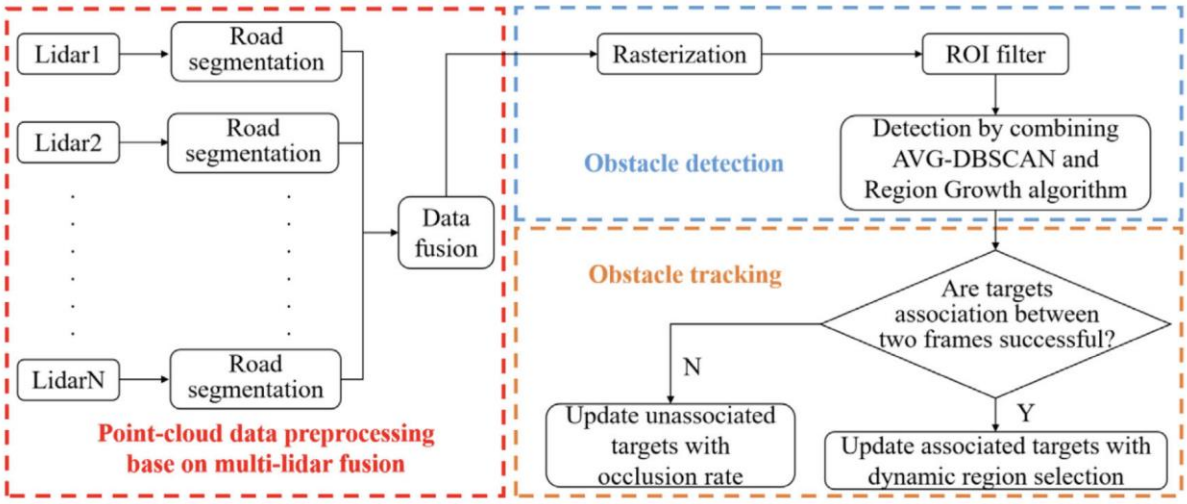
2.2.2 Data fusion models

Multiple data fusion algorithms have been used within the transport sector. In most of the literature, machine learning (ML), probabilistic, and time-series models have been employed in real-world scenarios. Additionally, neural networks (NN) and deep learning (DL) algorithms have been tested in both operational and non-operational scenarios. Data fusion applications are certainly moving towards the implementation of Intelligent Transportation Systems (ITS). According to the classification by (Meng *et al.*, 2020), the algorithms found in the literature can be organized into three categories: (i) signal-level data fusion, (ii) feature-level data fusion, and (iii) decision-level data fusion.

i. Signal level data fusion

Signal level is the lowest level of data fusion, where raw data serves as the input to the model, and the output is accurate and reliable data with minimal noises. The primary purpose of signal-level data fusion is to enhance the raw-data quality. These models are mainly applied in scenarios such as signal fusion, image fusion (or pixels fusion), and other similar applications.

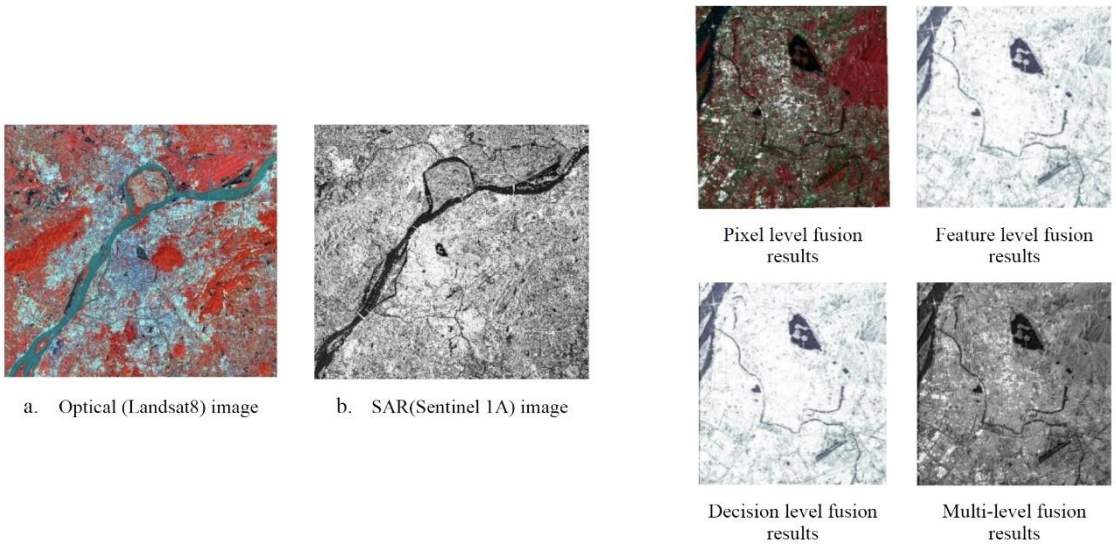
For estimating obstacle tracking in an **automated vehicle**, Li et al. (2021) uses motion compensation equations to fuse road segmentations collected by four VLP-16 lidars sensors installed separately in an experimental vehicle. As it is shown in Figure 5, the data fusion of the very initial collected and processed data, is the first step of their proposed method. Authors claimed that, by doing road segmentation fusion, the detection speed and accuracy improves to guarantee real-time performance (Li *et al.*, 2021).



Source: (Li *et al.*, 2021)

Figure 5. Obstacle tracking method proposed by Li et al. using raw data fusion

There are not so many applications in the transport sector on the signal level data fusion, however in literature I have found two applications that can be interesting if it is replicated in a transport application. Lin et al. (2019) have studied fusion methods of multi-source remote sensing images, they have fused optical images and aperture radar (SAR) images in pixel level, feature level and decision level. In Figure 6 is shown the original image versus the different DF results, authors claimed that the proposed method can significantly extract much information while maintaining relevant details of the original image (Lin *et al.*, 2019).



Source: (Lin et al., 2019)

Figure 6. (a) Original image sources vs. (b) Data fusion results

Similarly, there was tested different methods of pixel fusion for hyperspectral image and LIDAR data. Authors assessed wavelet transform, IHS transform and linear pixel fusion, they

concluded that linear pixel fusion is the best method as is the one that has lower computational time (Diana Abraham and Aravinth, 2018).

On the other hand, on the field of biometric deployments Drozdowski et al. (2022) have developed a pixel level fusion to propose a method of indexing biometric databases and retrieval protocol. Their results showed that computational workload can be reduced down to around 30% (Drozdowski *et al.*, 2022).

ii. Feature level data fusion

Feature-level data fusion enhances the representation of data, producing more polished and comprehensive information that presents various characteristics, features, or patterns of specific data inputs. The output from this fusion level can be utilized at the decision level or applied to other targets. As data input, it can use already processed data or previously extracted features. This level of fusion allows for more detailed analysis and can improve the performance of machine learning models.

Traditional methods for **traffic congestion detection** are often challenging to implement, costly, and potentially damaging to road surfaces. However, (Ke *et al.*, 2019) introduced an innovative approach that leverages the fusion of multidimensional visual features and convolutional neural networks (CNN) for detecting road congestion. This method assesses road congestion within a multidimensional feature space encompassing traffic density, traffic velocity, road occupancy, and traffic flow. Additionally, other methods for traffic detection have been proposed by using video content analysis (VCA); for example, (Lenatti *et al.*, 2023) developed a multi-camera VCA that alerts public transport drivers about riders, pedestrians and other vehicles detected by an SSD CNN as they approach the monitored area.

Various algorithms have been developed for predicting public **vehicle occupancy across networks**. For instance, (Ren *et al.*, 2020) proposed a hybrid integrated deep learning (HIDLST) model, combining Long Short-Term Memory (LSTM) networks and Residual Networks (ResNet), to predict citywide spatio-temporal taxi flow volumes. This model was tested in Beijing, China, over one week and the results indicated that HIDLST outperforms several existing deep learning models, demonstrating a significant advantage in dynamically capturing temporal dependencies in spatio-temporal flow volume series within a reasonable computation time.

On the other hand, Zhu et al. (2018) aimed to estimate **travel time** fusing data from three different sources: bus-based GPS, ILD and MPP. Furthermore, they explored its accuracy using as 'ground truth' the travel time estimated with automatic number plate recognition data (ANPR). They mainly used an ensemble algorithm; firstly, they classified the data into congested and uncongested traffic, using the *expectation maximization algorithm* to determine the performance of data fusion under different traffic-states. Then, they approached the data fusion in three ways: i. *Weighted mean approach* (WMA), ii. *Artificial Neural Networks* (ANNs) and iii. hybrid model WMA + ANNs. They concluded that the last one has the best performance, followed by the second and the first, respectively. Finally, they stated that the

final accuracy of travel time estimation depends on the reliability of individual data sources (Zhu *et al.*, 2018).

Another interesting implementation is the **user portraits definition**, a tool designed to construct a tagged user model based on users' behavioural habits and attribute characteristics. This enables the provision of diversified and differentiated services by offering a better understanding of bus passenger flow dynamics. For instance, (Zhu *et al.*, 2023) proposed a framework in which first GPS data, POI data and credit card data is fused to identify the tip chain to finally to use a K-Means algorithm to cluster the relevant features and define user portraits by identifying the features patterns over the different clusters.

These algorithms are utilized not only for estimation and prediction but also for feature importance analysis. For instance, the Light Gradient Boosted algorithm was employed to determine the significance of built environment and weather variables affecting daily ridership at the traffic analysis zone level, and to explore the non-linear relationships and interaction effects between these variables (Lin *et al.*, 2020).

iii. Decision level data fusion

Decision-level data fusion leverages the strengths of multiple decision-making processes. It represents the highest level of data fusion, presenting decisions for a task by integrating information that has already been generated. Each data source is processed independently, and the results or decisions from these independent analyses are then fused to produce a final decision. Decision-level methods often generate preliminary classifications and can combine different types of data to obtain accurate fusion results. This level of fusion is generally used right before final decisions are made.

For **transport network representation**, the authors have used a combination of recurrent neural networks (RRNs) and deep restricted Boltzmann machines (RBM) to model the traffic network. RNN has an internal state to provide a feedback loop from intermediate outputs, until the inputs layers; it allows a temporal sequence learning. RBM gives an extra loop between the visible and hidden layers. Then, authors proposed a traffic flow model based on convolutional neural networks (CNN) that considers inflow and outflow information in addition to traffic condition on the segment and helps to reduce the size of the feature maps because this algorithm was originally designed for image processing. A long-short term memory network (LSTM) to model traffic conditions was introduced as well; the author of this model affirmed that, to predict the short-term traffic condition for a network junction, the traffic patterns from its neighbouring roads can be used (Nguyen *et al.*, 2018).

Regarding **travel demand prediction**, deep learning techniques for short-term travel demand have emerged in order to exploit the big amounts of modern data sources, such as smart cards, sensors, GPS and CCTV. Accurate prediction in real time within a short computational time is the main goal. Among the methods used to reach this objective, authors did the comparison among featured-level data fusion model, deep neural network (DNN) and a stacked LSTM network; the results showed that the stacked LSTM network had the best accuracy among them. Furthermore, a fusion convolutional long short-term memory network

(FCL-Net) was proposed; it is a fusion model of multiple convolutional LSTM (conv-LSTM) layers, standard LSTM layers and convolutional layers. This model could provide better predictive performance because it captures the correlation of spatio-temporal features and external variables. Other deep learning (DL) models were proposed, a sequence learning model based on RNN and DNN forecast model. The first one was proposed to predict future taxi request using recent data and other external features, such as weather, time, and drop-offs. The DNN model was used to: forecast the demand for a mass rapid system considering explanatory variables, and for public transport demand prediction based on stop and stop-to-stop levels of demand, like dynamic OD estimation study (Nguyen *et al.*, 2018).

Concerning **traffic flow prediction**, various models have been developed, encompassing both traditional and non-traditional methods. Traditional methods include time series, Bayesian and probability models. For instance, (Pavlyuk, 2017) discussed statistical multivariate models such as ARIMA, VARMA and VECM for short-term traffic flow forecasting. Additionally, (van der Bijl *et al.*, 2022) compared different approaches for time series forecasting of motorway traffic flow rates at hourly and daily aggregation levels, testing TBATS, SARIMAX and LSTM models. Similarly, for estimating traffic demand in a junction, (Biletska *et al.*, 2009) proposed a multisensory data fusion approach for short-time OD matrix; they utilised four different method: the least square method with crisp[2] variables and coefficients (CLS), a linear programming (LP) problem with crisp variables and coefficients (CLP), an LP problem with variables and coefficients represented as intervals (ILP) and an LP problem with fuzzy variables and coefficients (FLP). The authors claimed that these approaches could address system inconsistencies and model data imperfections. Data collection was achieved through eight video cameras and counting sensors, providing all necessary information for estimation modelling. They concluded that CLP and FLP yielded the best results, while ILP was the least accurate.

In terms of non-traditional methods, two levels deep learning algorithms (DLA) was proposed, featuring a multi-output regression layer on the top and a deep belief network (DBN) at the bottom. The DBN is built as stacks of RBMs. The input space is connected to the DBN for unsupervised feature learning, followed by a sigmoid regression layer for supervised prediction. For a large traffic network prediction, the model operates on groups of streets clustered using a weighted clustering method, with the deepest layer of the DBN containing the most important and representative values. Experiments compared its performance with traditional methods, i.e. ARIMA, Bayesian model, support vector regression (SVR), artificial neural networks (ANNs), and the multivariate non-parametric regression model. The DLA algorithm excelled in testing road segments with low-to-medium levels of traffic flow but did not significantly outperform traditional methods for other levels of traffic flow.

Another DL algorithm was proposed for short-term traffic flow combines one combines a linear model trained with L1 regularization and a sequence of tanh layers. Its predictions, evaluated on data from special events with changed traffic flow regime, achieved a mean-squared error (MSE) 14% lower than a neural network with one hidden layer. Additionally, a

[2] Precise value

deep learning approach using a LSTM network was proposed. Based on experimental results, LSTM network outperformed all the other traditional methods, particularly for long-term forecasts (Nguyen *et al.*, 2018). Hybrid traffic prediction models that use both traditional and non-traditional methods have also been tested. For example, a combination of Prophet model and LSTM NN using VANETs data, or the combination of Markov chain and LSTM models using multiple road sensors (Pan *et al.*, 2024; Zaraket *et al.*, 2024).

In the literature can be found applications related to automated vehicles. Li et al. (2014) proposed a sensor-fusion drivable-region and lane-detection system for **autonomous vehicle navigation**. The authors presented feature-level fusion method for the LIDAR sensors and vision data, an optimal selection strategy for drivable-region detection and a conditional lane detection algorithm. For the features fusion they have tested three different supervised ML classification algorithms: artificial neural network (ANN), support vector machine (SVM) and adapting boosting (AdaBoost). Authors claimed that a combination of SVM and a genetic algorithm-based parameter optimization gave the best results with an accuracy of 88% (Li *et al.*, 2014).

Another application related to vehicle navigation is the one done by Sun et al. (2020) to estimate a high-quality vehicle state in urban environments. To achieve this, the authors have executed a two steps fusion using a robust extended Kalman filter (EKF) algorithm. First, the Global Positioning System (GPS) and monocular Visual Odometry (VO) were fused; and then, with the output of EKF_1, the Inertial Measurement Unit (IMU) was fused. The results show that the proposed IMU/GPS/VO fusion algorithm could deliver a 3D RMSE of 3.285m, which outperforms the other conventional candidate fusion schemes in the noisy GNSS urban areas. (Sun *et al.*, 2020). Likewise, Ramgopal et al. (2016) propose a multi-sensor data fusion technique applied to an automated bus rapid transport (BRT) system for vehicle localization using continuous landmarks. The data is collected by odometers and infrared distance sensors and processed using EKF method (Ramgopal and Manivannan, 2016). A further significant use case is the fusion of both spatial and temporal information, using RNN, for **forecasting bicycle flow** (Lu *et al.*, 2023).

Within the logistic applications, Yi et al. (2012), showed that multi-sensor data fusion could be used to predict **dangerous states** of hazardous chemicals' transport tank. The authors proposed data processing at different levels. In the first level, the weight average method was applied to smooth data and lessen the disturbance. Then, they estimated the signal slope trend using least-squares method and, finally, they evaluated the present value of the liquid level through a Kalman filter. Finally, based on the operation patterns and under different patterns, tank's dangerous state is obtained. (Chen *et al.*, 2012)

Furthermore, to improve the **decision making in emergency situations**, Chang et. Al (2014) measured reliable traffic speed using stationary vehicle detectors (VDs), space speed data collected by probe vehicles (PVs) and historical data. They proposed a framework to combine both datasets. It is divided in three sections: i. Data detection, ii. Data processing and iii. Fusion of results. Firstly, rush hour data is recollected to determine the weights in data processing, moving average and space mean speed as well are calculated. The weights were obtained based on the entropy of the data source. Then, after using *Shannon's Entropy* to

calculate the entropy, a *Lagrangian relaxation* was executed obtaining Eq. (1) which corresponds to the formula to calculate the weights. H_i corresponds to the entropy of i data source and W_i to the weight for the i data source. On the other hand, to reduce the error produced by those levels with zero entropy, the entropy must be adjusted using an adjustment factor.

$$W_i = \frac{1}{H_i^2 \sum_{i \in S} H_i^{-2}} \tag{1}$$

Finally, the fusion of results is done by Eq. (2), where V_i is the space mean speeds from data sources.

$$V_f = \frac{\sum_i W_i V_i}{\sum_i W_i} \tag{2}$$

To assess the accuracy of the framework, the authors evaluated it using the mean absolute percentage error (MAPE), Root mean square error (RMSE) and TheilU. The results indicated that the MAPE and TheilU values reflects a statistically good estimate of the fusion speed, while the RMSE suggests minimal fluctuation in the speed (Chang *et al.*, 2014). Similarly, efforts have been made to predict passenger volume under the influence of major public health emergencies by constructing a back propagation (BP) NN using public transport data, per capita disposable income data, and taxi data (Li and Xie, 2021).

The literature mention works related to **passenger flow prediction** models or OD matrices, important for the construction and design of a transport network. Also, for improving the operations and do adjustments in the system. Liu and Chen (2017) focused on passenger flow prediction model for improving the Mass Rapid Transit -MRT- a traffic tool from Taipei's public transport system. They built the model with DNN, using as input various combinations of factors such as historical passenger flow, temporal factors, directional factor, and holiday factor. The authors concluded that DNN captured the non-linear relationship between influential factors and passenger flows, and that those factors affect the performance of the prediction model (Liu and Chen, 2017). Similarly, (Gallo *et al.*, 2023) proposed a network-wide prediction framework for urban public transport vehicle occupancy using the LightGBM algorithm, supported by a behavioural model that correlates specific variables with occupancy. Testing this framework in Zurich, Switzerland, revealed that considering occupancy predictions for a target line while simultaneously accounting for other lines in the network significantly enhances prediction accuracy. This improvement is particularly notable in corridors served by multiple interacting lines.

Bachir et al. (2019) proposed infer dynamic origin-destination (OD) flows by transport modes using mobile network data. The authors claim that this would be the first methodology of this kind. They processed mobile network data and combined it with transport network geospatial data, travel survey, census, and travel card data. First, they extracted the categorical transport labels, i.e. road and rail, based on a combination of OpenStreetMap information and

the mobile network sectors. Then, they clustered the groups of sectors with similar transport usage through an agglomerative hierarchical clustering. Using a Bayesian inference, they determined the main transport mode associated to a mobile phone trajectory. And, once the mode is defined, the OD matrices were built in two levels of spatial aggregation in order to reduce the errors of the flows. These levels are departments and postcodes. The authors applied also an expansion factors to the OD matrices equal to the ratio of the total number of residents to the number of mobile subscribers of operator living in the same area. The travel survey and travel card data were used mainly for the validation. The authors concluded that this model can be reproduced if the same type of data sources is available, nevertheless is important to join mobile network data with travel surveys, public transport data and traffic counts to assure an effective calibration of OD flows (Bachir *et al.*, 2019).

To predict short-term bus passenger flow, (Mei *et al.*, 2023) proposed a traffic forecast model based on the attention mechanism (TFMA) which integrates various data sources, including AFC data, bus route information, bus schedules, weather conditions, and dates. By leveraging the real-time relevance of bus stops, the TFMA model aims to provide accurate and timely predictions of passenger flow. In contrast, for a long-term OD demand prediction, (Sobrino *et al.*, 2023) have use an XGBoost model with AFC data, station information and number of entries and exit per station.

To reduce complexity without compromising accuracy, (Coulaud *et al.*, 2022) proposed a L-shaped regression model to estimate one-station-ahead passenger flow using APC and AVL data. While (Shiao *et al.*, 2017) predicted the passenger flow of MRT in Taipei using a collection of various factors as inputs for a random forest model. The authors concluded that their random forest prediction model effectively captures the relationship between passenger flow and influential factors. They also found that the day of the week, direction and holidays can enhance the accuracy of prediction. The more detailed the data, the better the results.

Alternatively, for predicting the flow of crowds, (Hoang *et al.*, 2016) proposed a prediction framework that uses human mobility data, weather conditions and road network data and exploits multiple complex factors by decomposing the flow into seasonal, trend, and residual flows and modelling it with Intrinsic Gaussian Markov Random Fields (IGMRF). Similarly, (Toto *et al.*, 2016) proposed a framework for accurately predicting multi-granular arrival crowd flow at subway stations by extracting streaming features and station profile features from diverse urban data, including subway transaction data, weather data, and calendar data.

Another interesting application is related to the **passenger classification and trip purpose prediction**, in which multi-source data, i.e. AFC, travel survey, GPS and POI, is used to understand travel behaviours. Cluster, ensemble and ANN algorithms have been used under different scenarios and datasets showing valuable promising on the results to eventually improve and provide insights into transport planning (Sari Aslam *et al.*, 2021; Shi *et al.*, 2023; Zhao *et al.*, 2023).

2.3 Theoretical framework and literature gaps

The transport sector employs various methods for data collection, ranging from measurement sensors to communication packages, as well as categorization and prediction models, including traditional statistical models, ML and DL models. When focusing specifically on *passenger load estimation in public transport systems*, most efforts are directed towards optimizing demand-responsive bus services within the traditional fixed-route and fixed-schedule (FRFS) framework. Researchers aim to propose solutions that enhance the flexibility and responsiveness of the FRFS scheme (Filippi *et al.*, 2023). By optimizing passenger load estimation, services can be improved through schedule adjustments and real-time monitoring, enabling a more demand-responsive approach.

The literature reveals that various data types are typically used for passenger load estimation, including APC, AFC, AVL, time-related variables, weather related variables, geo-localization data, geospatial data, travel survey, census data and service schedule-related data are used for passenger load estimation; The size of the datasets can vary significantly, ranging from 700 records to 60 million records. Despite most of the data being available as open source, collaboration with transport companies is often necessary for studies, as APC, AFC, or AVL data are not always publicly accessible.

In terms of methods, data fusion for passenger load estimation employs statistical methods, ML, DL algorithms. The choice of method depends on the type of analysis, the complexity, and the level of interpretability required in the study. Statistical models rely on established statistical theories and principles to infer relationships between variables and make predictions. Machine learning broadens this approach by leveraging data-driven methods to enhance performance, while deep learning further extends this by utilizing complex neural networks to tackle highly intricate tasks that require large-scale data and computation.

DL models learn high-level features from the data, eliminating the need for domain expertise and extensive feature extraction. Conversely, for statistical models and some ML models, feature extraction is a process separate from the modelling itself (Mahapatra, 2018). This means that DL addresses problems end-to-end, whereas statistical models and ML typically break down the problem into smaller parts, which are solved separately and then combined.

Despite the promising results of DL for data fusion within the transport sector (Liu and Chen, 2017; Nguyen *et al.*, 2018; Li and Xie, 2021; Gallo *et al.*, 2023; Lu *et al.*, 2023; Mei *et al.*, 2023; Pan *et al.*, 2024; Zaraket *et al.*, 2024). it has several disadvantages. Firstly, DL models are hard to interpret, making it challenging to modify the model straightforwardly to achieve desired predictions. Secondly, DL requires significantly more data compared to ML algorithms and statistical models. Thirdly, the training phase of DL models is time-consuming and demands substantial computational power.

Ultimately, the most critical factor in building a model is the availability and quality of data. Without sufficient data, it is not possible to build an effective predictive model. The literature suggests that while DL techniques excel with large datasets, traditional statistical

models and ML algorithms are more effective when working with smaller datasets. (Mahapatra, 2018; Linford *et al.*, 2019)

As presented in Table 2, various statistical models mentioned in the literature for data fusion in passenger load estimation include Bayesian Inference (Bachir *et al.*, 2019), bi-autoregressive models(Coulaud *et al.*, 2022) and intrinsic Gaussian Markov random field models (Hoang *et al.*, 2016). For ML methods, the literature identifies LightGBM (Gallo *et al.*, 2023), KNN (Toto *et al.*, 2016), and ensemble algorithms like random forest and XGBoost (Toto *et al.*, 2016; Shiao *et al.*, 2017; Sobrino *et al.*, 2023).

Hybrid methods that utilize both statistical and ML models are also reported. These frameworks are typically structured in two sections: the first part involves estimating a missing feature, and the second part focuses on forecasting passenger load (Bachir *et al.*, 2019; Gallo *et al.*, 2023). Regarding DL models, the literature includes find DNN (Liu and Chen, 2017), ANN (Toto *et al.*, 2016), and encoders and decoder framework (Mei *et al.*, 2023).

Each of these methods presented in the literature claims to predict passenger load with low error. Additionally, the use of external variables is confirmed to positively influence the prediction model, suggesting that the combination of factors can increase prediction accuracy, including variables such as weather and built environment (Liu and Chen, 2017; Shiao *et al.*, 2017; Lin *et al.*, 2020; Gallo *et al.*, 2023; Mei *et al.*, 2023). However, it remains challenging to determine the best method for a specific dataset, as, to our knowledge, there are no studies comparing these models using the same dataset while considering dataset size and computational power required to produce the model.

On the other hand, various methods have been used for forecasting across different sectors, including the transport field, based on time series models. These include ARIMA (Cyril *et al.*, 2018; Ye *et al.*, 2019; F. Lu *et al.*, 2021; Wei *et al.*, 2022), SARIMA (Ruiz-Aguilar *et al.*, 2014; Andueza *et al.*, 2023; Munim *et al.*, 2023; Shrieenidhi M *et al.*, 2024), Facebook (FB) Prophet (Shakeel *et al.*, 2023; Ghosh and Jana, 2024; Wang *et al.*, 2024; Zaraket *et al.*, 2024) and Logit model (Yang *et al.*, 2022), all classified as statistical models. Additionally, Support Vector Machine (P. Wang *et al.*, 2021; Geetha *et al.*, 2024) and LSTM model (F. Lu *et al.*, 2021; Cheng *et al.*, 2024; Iyappan *et al.*, 2024; Yousuf et al., 2024), that are classified as ML and DL models, respectively.

A common feature among these methods is their good performance in forecasting highly complex and volatile variables such as stock markets, sales, energy consumption, electric vehicle aggregators, bulk volumes, heating consumption, and bus passenger demand and flow. Despite their good forecasting accuracy, there has been little exploration in the literature on using these techniques for data fusion in passenger load estimation. There is a significant potential for implementing and utilising these unexplored methods in this context.

From the literature, the algorithms that could potentially be more relevant for data fusion in passenger load estimation, including those used in other fields, are the following: Statistical Models: Bayesian regression, SARIMA, and FB Prophet; Machine Learning and Deep Learning Models: XGBoost, LightGBM, and LSTM models.

Table 2. Data fusion models for passenger load estimation

Estimation	Type of model	Datatypes used	Model used	Main application	Reference
Passenger flow prediction	DL - Neural Network	AFC Date Direction Day of the week Holiday	Deep Neural Network DNN	• Daily prediction • Taipei Metro • Experimental set up using only 1 station	(Liu and Chen, 2017)
	ML – Supervised Learning algorithm – Ensemble method	AFC Holiday Date Direction	Random forest	• Hourly prediction • Taipei Metro • Experimental set up using all the stations	(Shiao *et al.*, 2017)
	SM - Regression model	APC AVL	Bi-auto-regressive model	• One-station-ahead forecasting • Suburban railway network line of Greater Paris area • Experimental set up using 20 stations	(Coulaud *et al.*, 2022)
	SM + ML – Semi-supervised models	Mobile network data (Raw geolocation) Transport network geospatial data Travel survey Census data	Agglomerative hierarchical clustering(mobile network sectors) + Bayesian inference (determine main transport mode associated to trajectory)	• Daily OD prediction • Greater Paris area • Results by transport mode • Experimental set up	(Bachir *et al.*, 2019)
	ML – Supervised Learning algorithm – Ensemble method	AFC Num. entrances Num. exits	PDF+XGBoost	• Long term OD demand prediction • Madrid metro • Needed a sample generation model (PDF) • Experimental set up using 1 metro line	(Sobrino *et al.*, 2023)
	DL - Encoder-Decoder Architecture	AFC Bus route Bus stop information Weather Date	Traffic Forecasting Model based on Attention (TFMA)	• Real-time prediction • Suzhou public transport • The Attention mechanism calculates the real-time relevance between bus stops, allowing for adjustments to the predicted passenger flow based on changes in passenger traffic. • Experimental set up using 20 bus lines	(Mei *et al.*, 2023)

Continuation of Table 2. Data fusion models for passenger load estimation

Estimation	Type of model	Datatypes used	Model used	Main application	Reference
Crowd flows	SM – Stochastic processes	AVL City regions Weather Human mobility data Road network	Intrinsic Gaussian Markov random factors (IGMRF) model	• Hourly crowd flow • Citywide scale – results presented on cluster-based region grouping • Beijing and New York • Experimental set up	(Hoang *et al.*, 2016)
	SM , ML or DL – Model selection framework	AFC Time Weather Arrival and departure subway times Date	Prediction Framework For Usage Load on Subway SystEms (PULSE) Model selection: ARIMA, multilinear regression, K-Nearest, RandomForest, ANN	• Real-time framework • Shenzhen metro • Real-time arrival crowd flow prediction • It has a model selection in which use different models to predict and then rank them • Experimental set up	(Toto *et al.*, 2016)
Passenger occupancy	ML - Supervised Learning algorithm – Ensemble method	APC Schedule and real departure times Network variables Weather Holidays	Behavioural model least square-based + LightGMB	• Real-time, short-term and long-term predictors • Zurich public transport network • Considers other network lines when predicting a line's occupancy • Experimental set up using 1 tram line	(Gallo *et al.*, 2023)

- Statistical Models

Bayesian models refer to obtain a conditional probability, using Bayes' theorem, for a given data. Concerning its pros and cons, Bayesian methods work well when the models are flexible enough to capture relevant properties of the data, are coherent, conceptually straightforward, modular and have good performance. Furthermore, such methods work well with less training data. Nevertheless, the limitations are that they are subjective, it is hard to generate the prior probability of the model and they use approximations, which weaken the coherence of the argument. In most of the models it is expected that the features are independent, so sometimes they cannot be applicable in real life scenarios. Unlike other statistical models, Bayesian Ridge Regression occupies a grey area between statistical and machine learning models, sometimes called Statistical learning. This dual nature allows it to leverage strengths from both domains. (Ghahramani, 2012; Varghese, 2018; James *et al.*, 2021)

SARIMA model extends the ARIMA model to address seasonality in time series data. By incorporating seasonal differencing, SARIMA effectively captures and predicts regular seasonal patterns. This model consists of four components: (i) Seasonal (S): Adds seasonal terms to ARIMA to capture recurring patterns in the data; (ii) Autoregressive (AR): Utilizes the dependency between an observation and several lagged observations; (iii) Integrated (I): Applies differencing to raw observations to achieve stationarity in the time series; (iv) Moving Average (MA): Relies on the relationship between an observation and the residual error from a moving average model applied to lagged observations. SARIMA offers the advantages of rapid calculation and flexible modelling without sacrificing performance. However, it relies on historical data and struggles with capturing nonlinear characteristics. (Box *et al.*, 2015; Vagropoulos *et al.*, 2016; Mei *et al.*, 2023) When exogenous variables are included in the model the model is named SARIMAX and its general form is $SARIMAX(p,d,q)(P,D,Q,s)$ in which the parameters are defined as:

- **Non-seasonal parameters:**
  - p**:** Number of lag observations (AR part).
  - d**:** Number of differencing steps (I).
  - q**:** Size of the MA window.
- **Seasonal parameters:**
  - P**:** Seasonal AR terms.
  - D**:** Seasonal differencing steps (I).
  - Q**:** Seasonal MA terms.
  - s**:** Time steps in one seasonal period.

FB Prophet is a time series forecasting tool developed by Facebook. It incorporates an additive model that allows for non-linear trends to be combined with time-varying seasonal effects for weekly and daily data, along with the impact of holidays, thus the time series is decomposed on three components: trend, seasonality and holidays. Prophet is built to be user-

friendly, providing an intuitive interface for fitting and predicting time series data. Also, provide flexibility and robustness by handling missing values, outliers, and strong seasonal patterns. However, it offers less fine-tuning compared to other time series model and may struggle with very large datasets. FB prophet simplifies the forecasting process with its automated features, making it an good choice for quick and reliable forecasts without deep statistical expertise (Chaturvedi *et al.*, 2022; Munim *et al.*, 2023; Shakeel *et al.*, 2023).

- Machine Learning and Deep Learning models

For machine learning ensemble models are focused on, these combine decisions from multiple models to improve the overall performance (Ramzai, 2019); they train multiple learners to estimate the objective. Ordinary learning approaches try to construct one learner from training data, while ensemble methods try to construct a set of learners and combine them. Ensemble learning is also called committee-based learning or learning multiple classifier systems. There are two types of ensemble algorithms: sequential ensemble algorithms, or boosting models, where the base learners are generated sequentially, and, when the base learners are generated in parallel, they are called parallel ensemble methods, or bagging models (Zhou, 2012).

XGBoost is an advanced implementation of gradient boosting machines. It constructs an ensemble of decision trees sequentially, with each new tree built to correct the errors of the preceding ones, optimizing the overall model by minimizing a specified loss function. This model offers scalability, fast training speed, good performance and supports parallel processing; however, parameter adjustment can be complex, and it could consume a high amount of computational memory. Despite these challenges, XGBoost is ideal for structured data, as it can effectively handle large datasets and complex data structures. (Brownlee, 2016; Mei *et al.*, 2023)

LightGBM is a lightweight gradient boosting tree that utilizes advanced theoretical and technical optimizations, such as histogram-based sampling of features, to boost performance and accuracy while effectively reducing memory consumption and calculation costs. However, it can be sensitive to noise and overfitting, and requires the hyperparameters to be properly tuned to get the optimal performance. LightGBM can outperform XGBoost in terms of computational speed and memory consumption (Ke *et al.*, 2017a; Che, 2021; van Wyk, 2023)

In comparison from the previous algorithms, LSTM model is an implementation of improved RNN and is categorized as DL algorithm. It is designed to have improved memory management through linear memory cells, which are controlled by a set of gate units. These gate units regulate the flow of information, determining when information should enter the memory, when it should be forgotten, and when it should be outputted. Moreover, by not using any activation function, it prevents the vanishing gradient problem during backpropagation. Enabling it to possess robust data fitting capabilities, adaptability, and the capacity to handle large-scale datasets effectively (Swamynathan, 2019; Mei *et al.*, 2023).

Different transport data collection systems exist in the market and each of them adopts different technologies. Hence different data processing methods are used to estimate passenger load and mobility patterns. Bluetooth, Wi-Fi and cameras are part of the emerging technologies most used in the recent years in the transport sector, mainly in those applications which require estimation of real-time performance indicators such as load factor and boarding and alighting of passengers. Even though some APC systems claim to have an accuracy rate of 98% to 99% (Table 1), their performance in real-world operations ranges from 50% to 78% (Pronello and Garzón Ruiz, 2023). This discrepancy highlights a significant gap in comparative real-world evaluations, need to know in order to more accurately determine their performance under uniform conditions.

Despite the extensive application of both artificial intelligence models and statistical models in transport analytics, the literature provides inconclusive evidence regarding the superiority of one approach over the other in terms of results and application. Moreover, models used in other fields with strong potential for achieving high accuracy and performance in transport applications have yet to be explored, and the influence of dataset size on the decision-making process regarding the choice of modelling technique remains underexplored.

This research is aimed to understand how much a fused dataset can provide a better insight into the complexity of mobility patterns and to define guidelines to support transport companies in estimating on-board passenger count. This is achieved by collecting multiple sources data – including real operation bus data, meteorological and environmental data – implementing multiple data fusion algorithms to estimate vehicle load and evaluating the results not only in terms of performance but considering also the dataset size, computational power and other factors that could affect the implementation of these models in real scenarios in the transport sector.

Literature on the implementation and comparative analysis of different data fusion algorithms in the transport sector using manually collected operational data as ground truth is limited. Therefore, this study contributes to the field by collecting and analysing 57 days of real-operation data (collected over nine months to cover different periods and seasons) and implementing various statistical, machine learning, and deep learning models to estimate the number of passengers.

Addressing discrepancies between the claimed and actual performance of Automated Passenger Counting (APC) systems and exploring modelling process as a continuous cycle, where each model is compared with real-world data to evaluate its applicability and representation of reality is challenging.

Chapter 3

Objective and methodology

Considering the gaps identified in the literature, this research aims to conduct a thorough empirical examination of passenger count estimation models developed through both artificial intelligence (AI) and traditional statistical modelling techniques. The study seeks to compare these models in terms of performance, data requirements, and applicability, while also exploring the efficacy of data fusion methodologies in estimating the number on passengers on board.

The specific objectives of this thesis include:

- developing and implementing a set of passenger count estimation models using a various range of artificial intelligence techniques and statistical methods;
- exploring some of the estimation methods used in other fields and applying them to the transport sector, specifically for passenger load estimation;
- evaluating the models' performance through a comprehensive set of metrics to assess their accuracy and effectiveness in real-world scenarios;
- providing a comparative analysis of the applied models to identify their strengths and weaknesses in the context of passenger count estimations. This analysis will consider factors such as model complexity, applicability, scalability, and dataset size.

To reach the above objectives, an eight-step methodology has been defined:

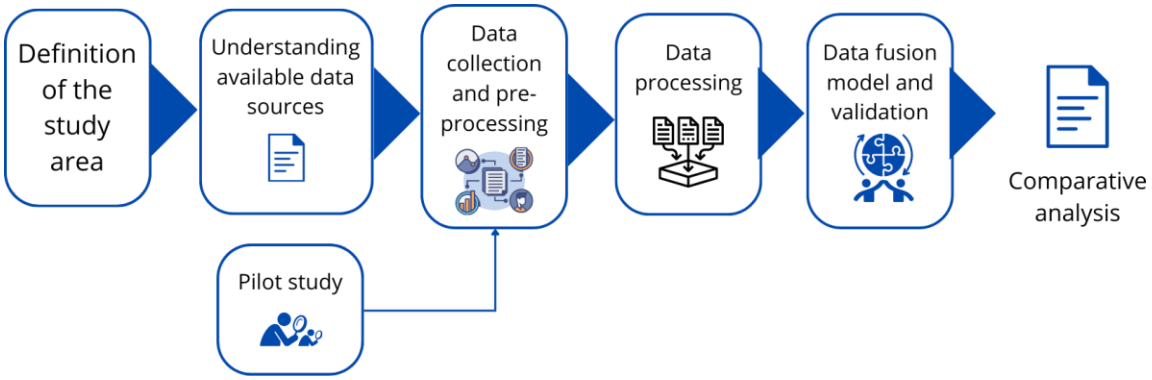


Figure 7. Methodology steps

I. **definition of the study area.** The pilot cities for the study are selected according to specific criteria involving geographical and demographic variables; transport supply; and technology adopted to count passengers (APC);

II. **understanding available data sources:** list all potential data sources that could provide information relevant to passenger counts. These may include, but are not limited to, automatic passenger counting (APC) systems installed in vehicles, Wi-Fi and Bluetooth tracking systems, camera footage, manual counts, survey data and territorial variables;

III. **pilot study:** a pilot study is set up to determine the feasibility of data collection and evaluation strategies on a smaller scale. This phase aims to identify potential challenges and refine the methodology before the full-scale study;

IV. **data collection and pre-processing:** the data collection of passengers boarded and alighted together with load factor involves a two-fold approach: collection of data from different APC systems and validation or ground-truth data that will be used to validate the APC systems. Since the external variables are coming from different data sources with different typologies, some pre-processing such as data cleaning and data imputation may be done. In this step each original format may be transformed into a standard format to simplify further processing**;**

V. **data processing:** in this stage, we focus on merging all datasets into a unified database. This involves integrating various file types by combining trip information between stops with available territorial variables, forming the foundation for our analysis. Significant efforts were made to prepare and refine the data, ensuring it is optimal for the study;

VI. **data fusion models and validation:** collected data is processed to estimate a single metric: vehicle load. Six different algorithms are used: three statistical models and three artificial intelligence models. This step involves describing how data fusion is performed, identifying important features for the models, establishing the validation metrics to assess the algorithms, and detailing the iterative process where these algorithms are tested and compared to identify the model that best fits reality. The outputs of the models are evaluated using a comprehensive set of metrics on the testing set;

VII. **comparative analysis**: this section uses Multiple Criteria Decision Analysis (MCDA) to compare artificial intelligence and statistical models, focusing on performance, data requirements, and scalability. The primary goal is to identify the most effective model among those studied. Additionally, the analysis aims to develop a strategy for transport companies that can be easily replicated to determine which model best fits specific situational requirements. By evaluating each model against these criteria, the best overall model is defined.

3.1 Definition of the study area

Mobility in urban areas is highly complex because of the great number of origins and destinations and the multiple modes used. In addition, transport systems are part of a dynamic system that is conditioned by exogenous and endogenous variables; the main components of urban dynamics are: land use, transport networks, people and vehicle flows, employment and workplaces, population and housing (Rodrigue, 2020). Thus, each city has its level of complexity according to the different territorial variables.

For the purposes of this study, we would like to test different combinations of factors from a single city to understand if these factors have an influence on the mobility prediction from the data fusion techniques. For this reason, and considering the data availability, the metropolitan area of Turin has been chosen for testing the models as it is a medium-large size city, with a rationally complex transport system.

The metropolitan area of Turin is situated in the Piedmont region of northwest Italy and covers a total area of approximately 6827 square kilometres, around 312 towns, with 2.3 million inhabitants. The main city, Turin, is positioned at the coordinates of 45.0703° N, 7.6869° E and covers a total area of approximately 130 square meters. Turin is notably located near the confluence of the Po and Dora Riparia rivers, with the majestic Alps framing its skyline to the northwest. This strategic location not only enriches its scenic beauty but also plays a significant role in the city's climate and environmental characteristics (*Città Metropolitana di Torino (CMT)*, 2024).

Turin has a population exceeding 800,000 residents, making it one of the largest cities in Italy (*Comune di Torino (TO)*, 2024). The population density reflects its urban nature with a diverse demographic composition. Over the last years, Turin has seen a significant change in its ethnic composition, now hosting a vibrant mix of Italian and international communities, including large groups from Romania, Morocco, and China, among others. This multicultural aspect is crucial for studies focusing on social integration and urban dynamics.

Historically, Turin has been a powerhouse of Italy's economy, significantly attributed to its automotive industry, with Fiat's foundation in the city marking a pivotal economic evolution. While still a vital sector, Turin's economy has diversified, embracing technology, education, and tourism. The city's universities and research institutions play a crucial role in

driving innovation, making it a fertile ground for studies on economic transition and industrial innovation.

Turin's transport infrastructure includes buses, trams, an underground metro line, and rail services, demonstrating commitment to sustainable transport efforts such as promoting cycling with bike line expansions and improving pedestrian zones in the city centre. The public transport system is managed by Gruppo Torinese Trasporti (GTT), which operates on the urban and suburban area with more than 900 vehicles serving about 91 routes (GTT - Gruppo Torinese Trasporti, 2022) covering the metropolitan area. Due to its urban dynamics, the metropolitan area of Turin provides significant insights into mobility patterns, characterised by its transport' system complexity. Furthermore, GTT has agreed to assist our study by offering data access and the opportunity to conduct experiments on their vehicles during operational hours. Consequently, Turin's metropolitan area is the primary focus of our research.

.

3.2 Understanding available data sources

In this section, the core of the research is explored by showcasing the data that serve as the foundation of the study. The advancements in technology and its accessibility have led to an unprecedented increase in sources of data, presenting both opportunities and challenges. For this reason, it is crucial to choose data sources that are not only relevant but also reliable for impactful research. Therefore, the sources used in this study are introduced and information about the origins, extent, and features of the data is shared.

3.2.1 Manual Passenger Count data.

Manual passenger count data was gathered to serve as 'ground truth data' throughout the study, providing a critical point of reference for evaluating the different models that estimates passenger count. This data were obtained through manual efforts, requiring a team of individuals to be present on-board during collection, which introduces the potential for human error. Consequently, determining the most effective method of collection was essential. Therefore, a pilot study was conducted before the data gathering phase to decide on the necessary information to collect, the most suitable data type, and the most efficient collection method. This study, along with the data collection, took place in Turin in collaboration with GTT. Due to technical difficulties encountered on GTT's part, the data collection process extended beyond the anticipated timeframe. Hence, in total were collected over 57 days spread over the period from November 2022 to July 2023.

3.2.2 Automatic Passenger Count systems

As detailed in the state-of-the-art section, regulations promote the use of Automatic Passenger Counting (APC) systems in public transport to enhance the system over the medium to long term. These systems are mandated for the purpose of counting the number of passengers using the transport system and subsequently of analysing the data to refine the network.

The availability of APC data for this research was made possible through a collaboration with GTT and Mobyforall Srl (MFA). MFA, is an innovative start-up and a technology and consulting firm, offers services and solutions to transport operators (for both passengers and freight), policymakers, and individual citizens by gathering and analysing data. They have developed a WiFi-based APC system installed on selected GTT buses, while GTT had previously implemented camera-based APC systems from different suppliers at every bus door. The data were specifically collected from buses equipped with both APC systems on buses designed with three doors, all of which are used for both boarding and alighting passengers, as depicted in *Figure 8*.



Source: (Chiara Appendino, 2019)

Figure 8. GTT bus.

Consequently, GTT provided passenger count estimates from the camera-based APC systems, while MFA supplied data from the WiFi-based APC system. In both instances, the collected data were transmitted to their respective internal clouds, and the processed information, including total passenger counts, was subsequently shared..

3.2.3 Environmental data

Turin has made certain environmental datasets publicly available through official platforms as open-source resources. Since environmental data can significantly influence the use of public transport (Lin *et al.*, 2020), these open-source datasets are essential for analysis and planning. Among these resources there are the data collected by the Regional Agency for Environmental Protection (ARPA) Piemonte. ARPA plays a key role in monitoring the environment, providing technical-scientific support and offering advice, also conducting other tasks beneficial to the region in adhering to legal mandates related to environmental prevention and protection (Arpa Piemonte, 2024). They gather meteorological, hydrological and pollution data from sensors distributed across the Piedmont region. The following environmental parameters can be obtained from the meteorological dataset[3]:

[3] https://www.arpa.piemonte.it/dati-ambientali/richiesta-dati-orari-meteorologici Accessed on: December 2023

1. **precipitation**: this is reported as the cumulative amount of precipitation measured over an hour, with the reported time marking the end of the measurement interval. The unit of measurement is millimetres (mm);
2. **air temperature**: temperatures are recorded at the end of each hour and are expressed in degrees Celsius (°C);
3. **wind speed**: wind speed is recorded at the end of each hour and is expressed in meters per second (m/s). For the conversion to kilometres per hour (km/h), wind speed is multiplied by 3.6, while for the conversion to knots, it is multiplied by 1.9;
4. **wind direction**: wind direction is measured in sexagesimal degrees, with 0 degrees representing north and then proceeding clockwise to 90 degrees (east), 180 degrees (south), and 270 degrees (west), completing the circle at 360 degrees (north again). Measurements are taken in an hourly basis;
5. **gusts**: the dataset records the maximum wind speed (in m/s) sampled in an hourly interval, with the time indicated as the end of the interval;
6. **air humidity**: this parameter represents the percentage ratio (%) between the water vapour in the air and the maximum amount of water vapour the air can contain at the current temperature and pressure. Measurements are taken at the end of the specified time interval;
7. **global solar radiation**: this includes both direct and diffuse radiation measured per unit of horizontal area, expressed in Watts per square meter (W/sqm). The data are sampled on an hourly basis;
8. **atmospheric pressure**: atmospheric pressure values are provided at the end of each hour and are expressed in hectopascals (hPa).

Additionally, ARPA has developed a platform named Aria[4] that provides air quality indicators such as ozone (O_3), nitrogen dioxide (NO_2) and particulate matter (PM_{10} and $PM_{2.5}$).

Details of meteorological sensors are provided in Table 3 and pollution sensors information is presented in Table 4, locations for ARPA sensors are presented in Figure 9.

Table 3. Meteorological data sensors - ARPA

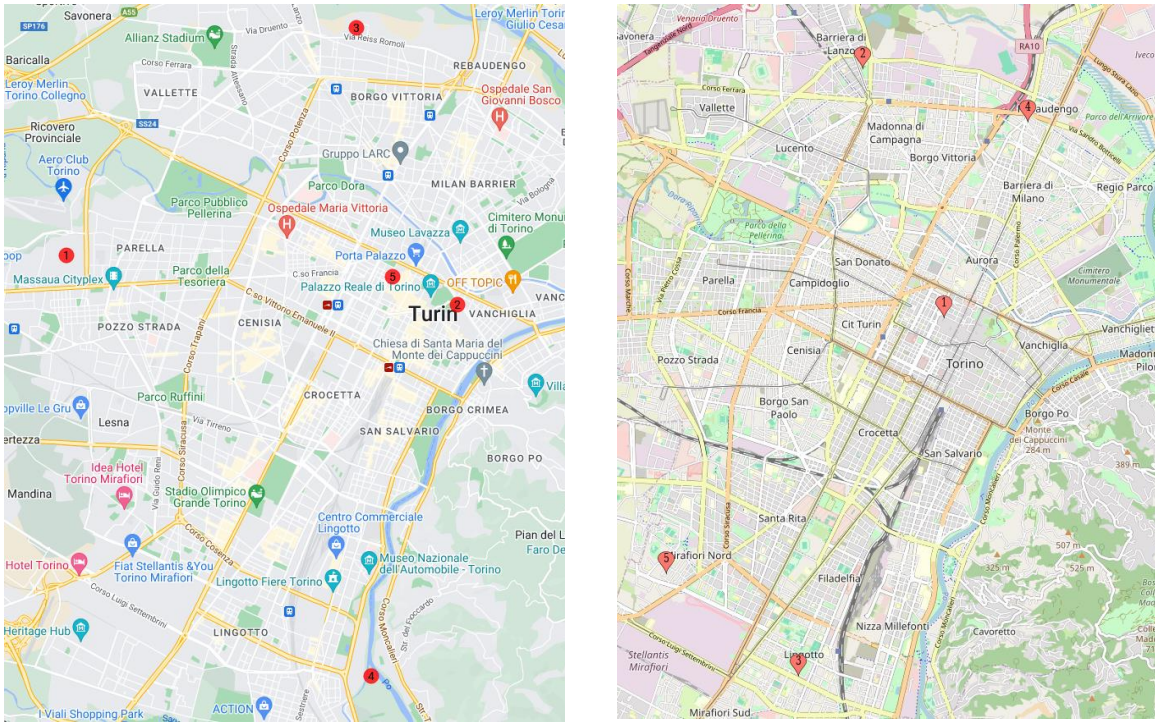
Sensor number	Name of sensor	Location	Province	Town	Quota (m s.l.m.)	Sensor	Facing towards	Starting operation date
1	TORINO ALENIA	CORSO MARCHE 41	TO	TORINO	280	Anemometro Igrometro Radiometro Termometro	N	6/1/2005

[4] https://aria.ambiente.piemonte.it/qualita-aria/dati Accessed on: December 2023

2	TORINO GIARDINI REALI	GIARDINI REALI	TO	TORINO	239	Anemometro Igrometro Pluviometro Radiometro Termometro	NNE	8/6/2004
3	TORINO REISS ROMOLI	TORINO	TO	TORINO	241	Anemometro Igrometro Pluviometro Radiometro Termometro	NNE	12/18/2003
4	TORINO VALLERE	VALLERE SMAT	TO	TORINO	239	Igrometro Pluviometro Radiometro Termometro	SW	5/18/2001
5	TORINO VIA DELLA CONSOLAT A	VIA DELLA CONSOLAT A 10	TO	TORINO	244	Anemometro Igrometro Pluviometro Radiometro Termometro	E	12/19/2003

Table 4. Pollution sensors location - aria ARPA

Station number	Station Name	Location	EOI code	Quota	Zone
1	Torino-Consolata	via della Consolata, 10	IT0469A	243m	urbana
2	Torino-Grassi	via P.Veronese ang. via Reiss Romoli c/o ITIS Grassi	IT0497A	245m	urbana
3	Torino-Lingotto	viale Augusto Monti, 21	IT2168A	243m	urbana
4	Torino-Rebaudengo	piazza Rebaudengo, 23	IT0470A	233m	urbana
5	Torino-Rubino	via Edoardo Rubino c/o giardini Rubino	IT1877A	257m	urbana



(a) (b)

Figure 9. Sensors location for (a) meteorological data and (b) pollution data

Another significant resource is GeoPortale Piemonte[5], a platform that provides a metadata catalogue of geographic information about the Piedmont territory. This catalogue allows users to search, access and download various formats and information related to territorial variables or the population distribution of Turin and its surroundings. Likewise, the statistical office of Turin, provides demographic information[6], serving as critical resource for research and analysis purposes. On the other hand, the open sources called Geofabrik[7] provide the option to download different point of interest (POI) from OpenStreetMap (OSM). From these sites, the registered population over Turin under different demographic variables and vectorial data of the buildings and POI, was considered. A sample of these data are shown in Figure 10.

[5] https://geoportale.igr.piemonte.it/cms/bdtre/modalita-di-pubblicazione-e-fruizione Accessed on: August 2023

[6] http://www.comune.torino.it/statistica/dati/demog.htm Accessed on: August 2023

[7] https://download.geofabrik.de/europe/italy/nord-ovest.html Accessed on: January 2024

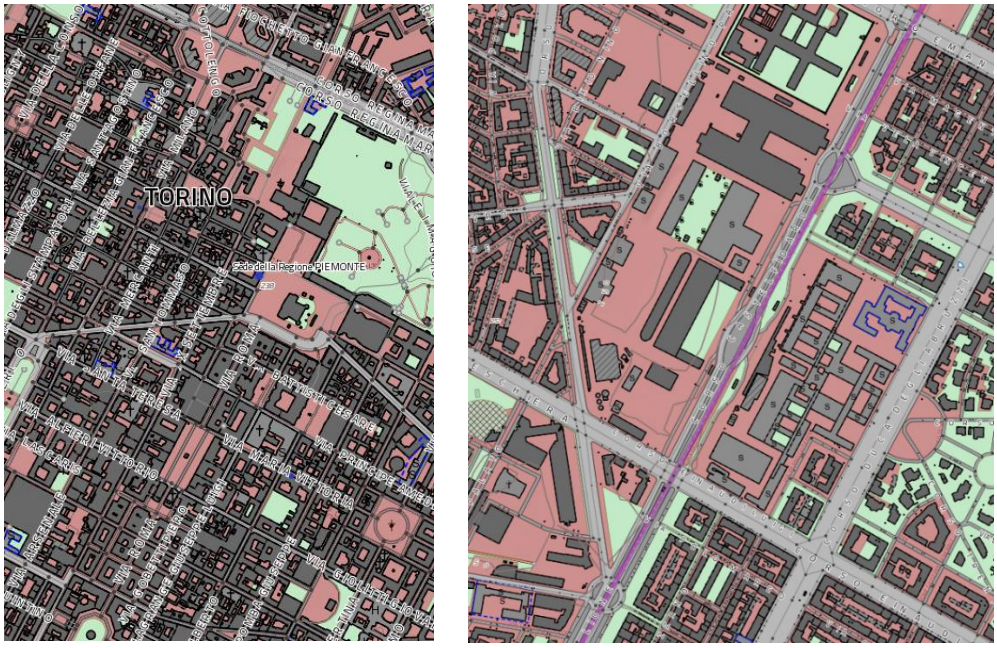


Figure 10. Vector shape for Turin buildings

3.3 Pilot study

The most important part of performing an experiment is to ensure that the execution of the experiment is going according to plans, as any errors in the experimental procedure could potentially nullify the experimental validity. For this reason, some trial runs should be executed to check the consistency of the experimental material, the measurement system, and the overall experimental technique. A lot of factors might be involved in the execution of an experiment. This pilot study involves understanding the technical implementation and processing of different APC systems, commercial and low-cost built by the author, and to assess their accuracy under real operational conditions.

To properly assess the accuracy of the commercial APC system and compare it with a low-cost system, a sufficiently large data base of ground truth data from the commercial APC system is needed. To this end, a methodology is designed, involving two in-field tests. First, the manual acquisition of ground truth data on a bus equipped with the commercial system is provided by Asti Servizi Pubblici (ASP), the public transport company in the city of Asti in the north-west of Italy. One month of ground truth data allowed the accuracy of the commercial system installed on the bus to be checked. Secondly, the comparison of the performance of the commercial system with that of a low-cost camera-based system was performed using a bus run by Gruppo Torinese Trasporti (GTT), the public transport company in the city of Turin, the capital of the region and situated 40 km from Asti. The commercial APC system is the same in the case of both companies, namely ASP and GTT: a camera-based system bought from a commercial supplier.

As summary, Figure 11 displays the methodology employed in the pilot study, which involved: a) the definition of available data sources; b) data collection on a bus, including the two in-field collections; c) selection of data processing methods; d) data analysis to support the design of a robust experiment for the main study.



Figure 11. Pilot study methodology

3.3.1 Definition of available data sources

The research was carried out in collaboration with three companies: GTT and ASP, which, respectively, oversee urban and suburban transport in Turin and Asti, and Mobyforall S.r.l., a startup that provides solutions and services to enhance transport systems. The available data sources include (i) a manual count collection that is needed for validation. Moreover, all companies facilitated the use of their respective APC systems, with GTT and ASP contributing data from (ii) a conventional camera-based APC that was already integrated into their vehicles, while Mobyforall provided (iii) a WiFi-based APC system. Furthermore, the literature review revealed that Deep Learning (DL) algorithms have been employed to attain precise outcomes in object recognition. Consequently, (iv) a low-cost camera-based APC system was developed for this study.

i) Manual count collection:

A campaign to collect data manually was designed to have reliable ground-true data to compare with those counted by the automated systems.

ii) Conventional camera-based APC

The conventional camera-based APC correspond to a stereoscopic system produced by Eurotech, that collects data of the passengers boarding and alighting on each door. This data was collected for both Asti and Turin.

iii) Wi-Fi-based APC system

The Wi-Fi-based data is introduced as it is provided by Mobyforall Srl (MFA)[8]. It collects Wi-Fi probe request packets broadcasted by smartphones in an anonymised manner to ensure

[8] Mobyforall (MFA) is a start-up based in Turin (Italy) and based at I3P, a start-up incubator inside Politecnico di Torino. MFA is a mobility technology company working on solutions for transport services that has developed a Wi-Fi based data collection system to count passengers.

privacy of smartphone users. MFA agreed to make this collected data available to the author for the purpose of this thesis.

iv) Low-cost camera-based APC system.

A low-cost APC system is a solution for counting passengers on public transport that is designed to be more affordable compared to traditional APC systems. In recent years, the interest of developing low-cost APC systems has increased because they can provide accurate passenger counts to enhance the overall understanding of passenger behaviour, in a cost-effective way. The low-cost system designed in this research is a camera-based system, as object detection algorithms have been extensively studied in recent years and are considered the most suitable technology for the task.

3.3.2 Design of low-cost APC system

The APC system was designed with a primary emphasis on achieving both a low price and good performance. The system developed was composed of the hardware, an object recognition algorithm, and an object tracking algorithm. Figure 12 shows the procedure followed in the design step, including the choice of a camera and a system architecture suitable for our experimental purposes. An initial object detection algorithm was built using data collected in a controlled environment, and this algorithm was subsequently refined using real application data collected in collaboration with GTT on one of their buses.

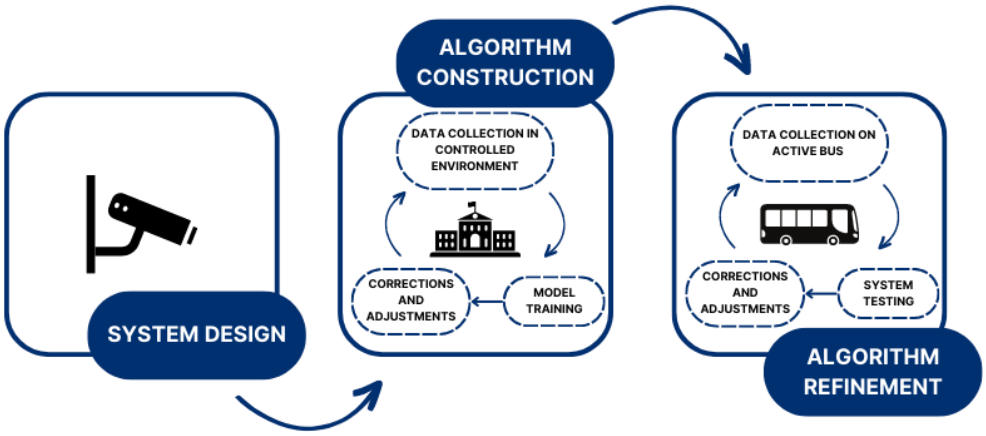


Figure 12. Methodological framework for the low-cost APC system

System design

As any APC system, this system is composed of both hardware and software; the physical APC camera is part of the hardware specifications, but how it works and where it is interconnected is related to the software. The system architecture is presented on Figure 13: it has a video recording system that connects with the backend system. In the backend, the detection algorithms would do the passenger detection and would estimate the number of passengers boarding and alighting in each door. To finally, provide to the final customer, i.e. transport company, the estimated data through a graphical user interface (GUI).

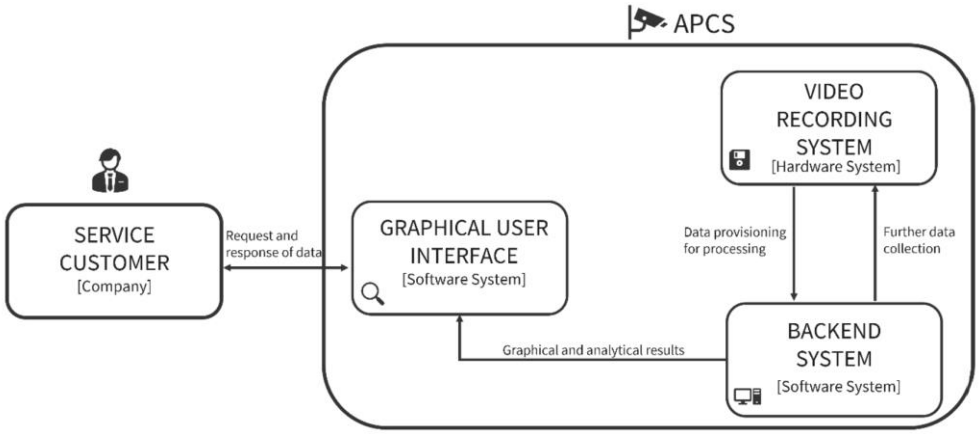


Figure 13. System architecture

The main components of the system were the board and the camera. The board was what connected the camera with the video recording and backend system. It also saved the recorded video in its internal memory and, if needed, was capable of running a real-time passenger detection algorithm. The board must, therefore, be powerful enough to handle these operations without any memory or heating issues arising. The camera needs to be compatible with the board system and to be able to produce good-quality images.

To define the board and the camera, there is a set of criterions to be followed that are based on some functional and technical requirements. These are:

- **good trade-off between price and accuracy:** there are already some technologies on the market that works similarly, but these are very expensive. In fact, the main brands that develop this kind of systems are Hella Aglaia, Eurotech and Retail Sensing and each system can cost thousands of Euros. Even though these companies claimed an accuracy of 99%, they cannot assure this result in a real scenario application;
- **vibration insensitiveness:** since the test is going to be done in a bus, this is an important requirement as they are exposed to intense vibrations due to road conditions. Especially in cities like Turin, the roads are very different between each other and different kind of surfaces like road, pavè and rails, can be found. These can lead to experience big vibrations on the vehicle and so the system is more likely to fell;
- **not invasive:** passengers should not feel intimidated by the system when they board the vehicle, so it is not convenient to place a camera right in front of each door as it will cause some discomfort. The equipment should be small, easy to manipulate and easy to be placed inside the vehicle. Since there is a possibility of being stolen by some malicious users, it should be located in a way that cannot be easily removed and, in a position, where it is not directly visible by the passengers when boarding the bus;
- **calculation capabilities:** the system should have calculation capabilities and good computational power to run some scripts. This requirement is important because it

will allow the camera to be easily controlled from remote, and the videos can be processed directly from it. Other important aspect is to have the feature of assigning the videos a custom name, which makes the processing phase easier, and to have the possibility to process video in real time;

- **30 FPS videos and a good resolution:** the available object recognition algorithms that work with videos of 30 frames per second. In addition, good resolution of the camera should be ensured in order to have the best possible performance in the detection algorithm;
- **system independent from the bus:** GTT did not authorized an intrusive installation of the system, thus it cannot be directly attached to the power line of the bus as this would mean to drill the wall for installation. In brief, the system as to be as independent as possible from the bus.

The Raspberry Pi 3 Model B was a suitable "universal" board, allowing any of the three cameras considered to be connected without any technical issues. It comprises a quad-core 1.2 GHz Broadcom BCM2837 64-bit CPU, 1 GB RAM, a CSI camera port, and a Micro SD port for extra memory space. Its operating system is Raspberry Pi OS (previously called Raspbian), a Linux-based OS including basic programs, utilities, numerous packages, and pre-compiled software (Turnip *et al.*, 2020). Raspberry Pi is, thus, a mini-PC with a computational capacity enabling it to execute Python scripts and lightweight object recognition models (Islam *et al.*, 2020; Kamath and A., 2021; Dhou *et al.*, 2022; Gharge *et al.*, 2023), making it a good choice as the system motherboard.

Three different cameras were considered: an SPI camera, a Raspberry Pi camera, and a HIMAX CMOS HM0190. The Raspberry Pi camera offered the best resolution, with the possibility of changing and regulating the resolution if needed. The image resolution, frame rate and board worked well together as a system, providing a good level of accuracy (Redmon and Farhadi, 2018; Anand and Kumawat, 2021; Jabbar *et al.*, 2021), and this is why the Raspberry Pi camera was chosen. Figure 14 shows the low-cost APC system assembled



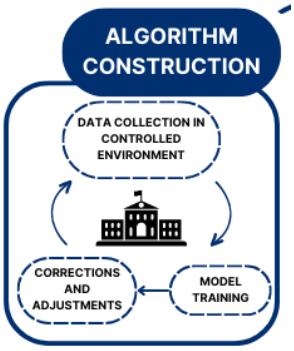
Figure 14. Low-cost APC system

Camera Algorithm construction

This APC system uses video camera as an algorithm input to estimate boarding and alighting of the users. Hence, these frames make up the input dataset for implementing deep learning techniques, mainly object detection and tracking algorithms.

To achieve image recognition, it is needed machine vision technologies in combination with a camera and artificial intelligence software. Object recognition model can identify objects, places, people, writing and actions. It identifies all relevant objects in an image or video; while object detection models identify all the occurrences of a specific, and only, object in a given image or video. For this pilot study, object detection algorithms will be used since this model aim to detect all the occurrences "person" within a set of photos.

Nevertheless, to follow the path of a passenger and understand if it is boarding or alighting, a tracking algorithm is needed. The overall system should be able to keep track of a passenger across the sequential set of frames which compose the video, and to infer his trajectory. Finally, a counting criterion must be chosen to memorize the number of boarded and alighted passengers along an entire vehicle trip.



Source: (Pronello and Garzón Ruiz, 2023)

Figure 15. Algorithm construction steps

Source: (Pronello and Garzón Ruiz, 2023)

Figure 15 presents the steps related to the algorithm construction. First, data is collected in a controlled environment. Then, a first model is trained, tested and later calibrated based on corrections and adjustments.

1. Data collection in controlled environment

Since this experiment was performed in a real setting, it was important to set up and test the system in a controlled environment before going in-field, in order to know the op-timal conditions for its deployment. The most significant factors are (1) the height at which the camera is placed, as this can cause detection issues; (2) the camera angle, essential for ensuring passenger privacy; and (3) the lighting conditions. The controlled experiment plays a crucial

role in highlighting any weak points, and it is essential that the simulated conditions reflect as far as possible the real in-field conditions. To this end, the controlled experiment included the following:

- The simulation of different lighting conditions;
- the simulation of different levels of crowdedness;
- the simulation of people's different boarding and alighting patterns;
- the simulation of people wearing and carrying accessories.

The controlled experiment was performed using a door at the entrance to a study room inside the main campus of the Politecnico di Torino. The door provides access from the exterior, so it was possible to obtain different lighting conditions, and moreover it is a door used by a large number of students wearing or carrying a variety of accessories. Entering the study room was considered to be 'boarding', and exiting was considered to be 'alighting'. In total, 10 h of data were collected, 70% of which was used as input for training the algorithm, and the remaining 30% for testing it. To this end, a pre-processing step was necessary to extract the frames to be used for the input dataset. The training set of the frames was then labelled, and the weights were trained.

2. Model implementation and refinement

On the literature there are different types of object detection and object tracking algorithms. Any of them focus on specific vision-related issues, i.e. lighting, and works with different pixel technologies and processing. Thus, the main purpose of this step is to define the best performance algorithm to be later used in the real-context application.

The following step concerned the object detection algorithm. YOLO is a one-step detector which has the advantages of high accuracy, good learning capabilities and a speed that allows real-time applications; it is three orders of magnitude faster than R-CNN while exhibiting good object detection (Soviany and Ionescu, 2018), and the literature indicates that one-step detectors represent a good compromise between object detection performance and computation time (*YOLOv5 | PyTorch*, 2022). In this section we trained different versions of YOLO and evaluated them with the accuracy, precision and recall metrics that are calculated using Equations (3), (4) and (5) respectively.

$$accuracy = \frac{TP}{TP + FP + FN} \tag{3}$$

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$recall = \frac{TP}{TP + FN} \tag{5}$$

where:

- TP is True Positives
- FP is False Positives
- FN is False Negatives.

Next came a step relating to the tracking and counting stage, for which the DeepSORT algorithm was considered the best option, as it is a multi-object tracking algorithm widely used in the transport sector and has been shown to have good accuracy (Verma and Verma, 2020; Pei *et al.*, 2021; Zhong *et al.*, 2022). The tracking and counting algorithm do not require a training phase, since it uses the pre-trained weights of the previous step as the input. The extent to which the tracking phase is successful is highly dependent on the confidence level achieved in the detection of each subject, avoiding ID losses and re-associations with a person already detected in the previous frame. The tracking phase leads on to the actual counting implementation, which consists of drawing two regions of interest (ROI) over the frame: the width and location of the ROI are determined according to the area of the frame where the tracking results are the most effective.

For our low-cost APC system to work as intended in a real setting, a refinement step was necessary; in-field data were collected to re-train the algorithm.

3.3.3 Data collection for pilot study

The performance accuracy of the commercial APC system is measured against ground truth in regard to three different features: boarding, alighting and vehicle occupancy. *The initial accuracy assessment* is performed in the city of Asti, which has about 75,000 inhabitants, with the collaboration of ASP, the local public transport company that runs a fleet of 64 vehicles, serving 23 urban and suburban bus lines.

Data is collected over a four-week period from 19 October 2021 to 19 November 2021, excluding weekends, from 7 a.m. to 4 p.m. Thus, 20 days are monitored, collecting approximately 180 h of data. The bus has two doors, and the passengers could board or alight via either of them. Four employees of the start-up Mobyforall (The startup Mobyforall - www.mobyforall.com, accessed on 13 September 2021- kindly provided the data collected by their team for this study) ensured an accurate collection of ground truth data; they were present on board the bus, two for each door. Time stamps, vehicle occupancy, and the number of passengers boarding and alighting at each stop were recorded using pen and paper. Because of certain planning and logistical constraints to which ASP was subjected, the data collection was carried out on a single bus, which is assigned to different lines during the collection period.

ASP then provided the numbers of boarding and alighting passengers as recorded by their APC system, in the form of a database in which each row contained the street name of the stop, along with the total count of passengers recorded as boarding and alighting at that specific stop. Since occupancy information was not included in the provided data, this was calculated during the subsequent data processing phase.

When *comparing the performance of the commercial system with that of a low-cost camera-based system*, different requirements had to be met to ensure the dataset heterogeneity necessary for preventing irregular behaviours of the algorithms and avoiding overfitting and underfitting. The collected dataset needed to be statistically representative of the population under study. We, therefore, paid particular attention to how training data for the in-field scenario were acquired, and used a sampling plan to ensure its representativeness. The formula shown in Equation (6) was used to calculate the required sample size, according to the literature (Verma and Verma, 2020):

$$n = \frac{(z)^2 \times p(1-p)}{\Delta^2} \tag{6}$$

where n is the sample size; z is the level of confidence according to the standard normal distribution (for a level of confidence of 95%, z = 1.96, for a level of confidence of 99%, z = 2.575); p is the estimated proportion of the population that presents the selected characteristic; and Δ is the accepted error (accepted error percentage, not exceeding 5%). In our case, the sampling unit is the passenger and p is set to 0.5.

Given the accepted error and the corresponding sample size, it was a question of determining how many days of recordings were needed to achieve the desired results. To this end, the formulas shown in Equations (7) and (8) were used:

$$n = p_{ride} \times ride_{day} \times days \tag{7}$$

$$days = \frac{n}{p_{ride} \times ride_{day}} \tag{8}$$

where n is the sample size; $p_{ride}$is the average number of people recorded during one journey; $ride_{day}$ is the number of journeys considered in one day of recordings; and days (the target variable to be determined) is the number of days needed for data collection.

The level of error selected was 2%, requiring us to count more than 2400 passengers. Since an average of 60 people (p_{ride}= 60) are transported per journey and considering four journeys per day by applying the formula Equation (8), we collected data over 10 days in order to achieve representativeness. As each journey takes around one hour, the total number of recording hours obtained was 40 h, corresponding to more than 5000 videoclips of data. The frames from these clips were labelled and used for re-training the weights in the algorithms.

The test was carried out aboard a bus belonging to GTT. Due to constraints related to planning and the availability of GTT personnel, the trial was performed in a single bus and involved manually counting passengers boarding and alighting over three days in 2022: 13, 30, and 31 May. Following the analysis of preliminary results, it was decided that three additional days of data would be required for a better understanding, and this was collected on 3, 7, and

9 November. We were, thus, able to compare passenger numbers obtained by the commercial APC, by our low-cost system, and by our manual count (the ground truth).

The bus assigned to the experiment has three doors that passengers can use for both boarding and alighting. The commercial APC system already installed on the bus by the transport company is a stereoscopic camera that counts the number of passengers boarding and alighting at each bus stop. Our low-cost device is small enough to be classified as non-invasive. A camera was located above each door so as not to point directly into passengers' faces (Figure 16). The passenger privacy stipulated by the company was, thus, ensured, since facial images were not recorded (Figure 17). The camera was activated manually to record a video every time the door was opened.



Figure 16. Low-cost APC installation

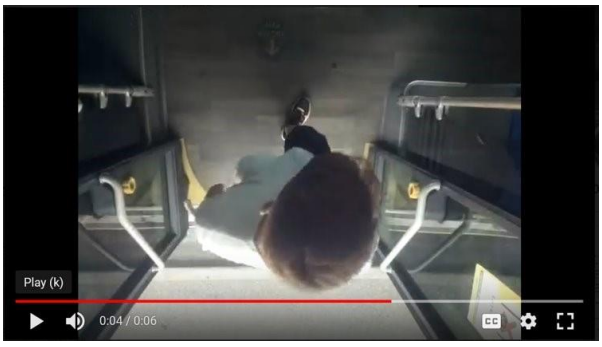


Figure 17. Low-cost APC camera video frame

The manual counting is collected by 6 manual counters – two for each door of the bus using the format specified in Table 5. The data is recorded manually.

Table 5 Pilot study - Manual data collection structure - fields and description

Field name	Description	Variable's type
Date	Date of data collection	Date
Time	Time in hours and minutes	Time
Line	Line on which the bus is travelling	Discrete quantitative
Direction	Forward or return direction	String
Stop name	Name of the stop	String
Door Position	Front, Middle or Back door	Nominal
Vehicle trip	The run number of the vehicle from source destination on a specific route/line. "Corsa" in Italian.	Discrete quantitative
In_count	Manual count of passengers boarding the bus through the door	Discrete quantitative
Out_count	Manual count of passengers boarding the bus through the door	Discrete quantitative
Vehicle load	Vehicle load on the bus after the bus leaves the stop	Discrete quantitative

3.3.4 Data processing for pilot study

On this step, each APC system is processed individually; each raw data is processed into a more usable and meaningful form for analysis, interpretation, and presentation. The goal of this step is to extract useful insights, knowledge, and information from data. Moreover, in this step, the vehicle load and the boarding and alighting of passengers are estimated for further analysis and comparison.

For *the initial accuracy assessment*, the data from manual counting and from the commercial APC were processed in order to enable a valid comparison. Data from manual counting, in the form of a sheet for each door, were digitalised and standardised to a single timestamp format; Data from the commercial APC only provided information on passengers boarding and alighting at each stop. Any lack of accuracy in this counting causes errors in the calculation of the occupancy between stops. These errors are cumulative, and the overall vehicle occupancy error will increase as a bus continues its journey. To handle this issue, some transport companies have opted to reset the vehicle occupancy at the end of each journey. We calculated vehicle occupancy using two different reset strategies: (i) vehicle occupancy was set to 0 at the beginning of each day; (ii) vehicle occupancy was set to 0 at the start of each journey, taking no account of passengers boarding or alighting at the terminus, because passengers often enter and leave a bus multiple times while it is at the terminus.

For the *comparison of the performance of the commercial system with the low-cost camera-based device*, the manual count and commercial APC processing were very similar to those used during the initial assessment. However, vehicle occupancy had already been estimated from the data provided by GTT, and some cleaning and imputation were done, since there were

empty fields relating to vehicle occupancy, and some stop-related information needed to be filled in.

The low-cost system required the use of the algorithm to estimate the different values. The frames of the videos at each stop were used to initially estimate the number of passengers boarding and alighting. The vehicle occupancy was then calculated for the testing days. Figure 18 shows the output of the algorithm; the person is identified and assigned to a centroid that is tracked over the video. The ROI lines indicated the direction of the passenger, that is to say, whether they were boarding or alighting.

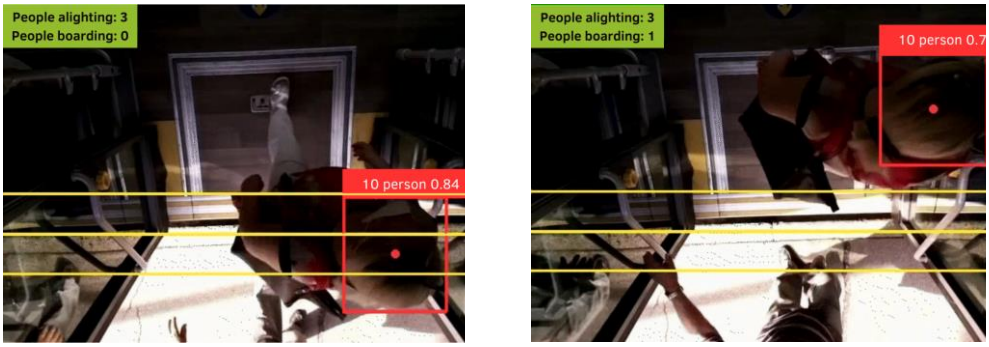


Figure 18. Output of the algorithm in the low-cost APC.

3.3.5 Data analysis and validation for pilot study

Different evaluation metrics were used to quantify the performance of the commercial APC system; these metrics express the accuracy of estimations with respect to the ground truth collected during the field study. The aim is to identify the accuracy in terms of boarding, alighting and vehicle occupancy.

To the best of our knowledge, transport companies generally report accuracies in terms of percentages (Asti Servizi Pubblici S.p.A., 2021; GTT - Gruppo Torinese Trasporti, 2022; Azienda Trasporti Milanesi, 2023), and thus, a choice was made to calculate the error by using the symmetric mean absolute percentage error (SMAPE), since this has an upper bound of 100% and a lower bound of 0%. In addition, this metric has recently been gaining more popularity, along with the coefficient of determination, within the machine learning community (Chicco *et al.*, 2021). The error was, therefore, calculated using the SMAPE formula reported in Equation (9):

$$SMAPE = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{\lceil A_t \rceil + |F_t|} \quad (9)$$

where n is the total number of observations; A_t is the actual value or the ground truth value at observation t; and F_t is the forecasted value at observation t.

The accuracy is then calculated using Equation (10):

$$Accuracy = 100\% - SMAPE \quad (10)$$

Conclusions are drawn to do not repeat the same mistakes for the data fusion experiment.

3.4 Data collection and data pre-processing

Data fusion experiment followed a different collection methodology in comparison with the pilot study as the methodology was improved from the experience of the pilot study. Additionally, the methodology was also discussed behind data collection and the criteria for inclusion, providing a clear view of the groundwork that supports our analysis. All the data sources have a different initial format. Hence, each of them would need a different treatment to reach the general one.

3.4.1 Manual Passenger Count data

The collection of these data was based on a sampling plan to make sure to have the significant representativeness required to cover the potential fluctuations of passengers during the peak and off-peak hours. Based on the timeframes defined by the Piedmont Mobility Agency (AMP), the peak hours in the public transport are: from 8:00 to 9:59 in the morning: form 13:00 to 14:59 when schools are finishing; and in the evening, from 18:00 to 19:59. There is not a specification of different peak hours for the weekends (AMP, 2013), so we defined the weekend peak hours to be from 16 to 19 looking at the data collected in the previous travel surveys AMP did.

The sample size was calculated using the Equation (6) presented in the previous section. Given an acceptance error equal to 5% and the interval of confidence of 95%, the number of days of collection to achieve the desired results were calculated. For this purpose, the formula shown in Equation (11) was used:

$$days = \frac{n}{ride_{day}} \quad (11)$$

where n is the sample size; $ride_{day}$ is the number of trips considered in one day of collection; and $days$ (the target variable to be determined) is the number of days needed for data collection.

The level of error selected was 5%, requiring us to collect at least 384 trips in total. Since 1 trip duration is 1 hour, in one day we could potentially do 6 hours of collections, and

considering 6 trips per day by applying the formula of Equation (11), 64 days of data are needed to be collected to achieve representativeness. However, due to multiple problems (service disruption, buses in panne, etc.) faced during the collection, the actual total number of days collected were 57 days distributed over weekdays and weekends, as reported in Table 6.

Table 6. Manual passenger count's days collected

Month	Number of days
Nov – 2022	3
Jan	1
Feb	10
Mar	5
Apr	5
May	15
Jun	17
Jul	1
Total collected days	**57**

The manual count started on November 2022 and was conducted on a single bus (2437) but multiple issues arose during the data collection: (i) the bus frequently returned to the depot, (ii) the lack of a fixed schedule in the mornings or the day before made it difficult to locate the bus daily, external tools like Telegram GTT bot, MFA mobile app, Moovit[9], and real-time location feature on GTT webpage were used to locate the bus, and (iii) there were instances where drivers, pressed for time, asked passengers, including the counting team, to leave the bus, resulting in further delays and lost time. Despite having a direct authorisation from GTT to use a camera to support the manual count, on multiple occasions (iv) the driver refused to allow the camera connection, making the manual count impossible. After discussing these problems, GTT provided two additional buses for the manual count from May 2023, increasing the total number of buses to three. This allowed for a more consistent count, as at least one bus was always operational. Despite this improvement, locating the buses daily remained challenging.

The vehicle's assigned for the experiment are presented in Table 7. The initial bus provided by GTT was bus number 2437 and then bus number 2402 and 2428 were made available for the data collection from May of 2023 GTT.

[9] Moovit is a real time public transit app that offers route planning and lines information

Table 7. Vehicles used for the study

Vehicle number	vehicleID for the study
2437	01
2402	02
2428	03

Data fusion experiment followed a different manual count data collection methodology, refined according to the experience of the pilot study. The team was composed by two data collectors with the support of a camera that was positioned on the inside ceiling of the bus, at the height of the central door once the daily counting started, and removed once the daily counting was over. The collector n. 1 was collecting data from both front and middle door simultaneously, while collector n. 2 was collecting data from the back door.

For this data collection two "mobile surveys" were used, one for each data collector, created using ArcGIS survey123, a simple and intuitive form-centric data gathering solution; a sample is shown in Figure 19. The first survey gathered the variables presented in

Table *8* for the front and middle doors, while the second survey collected the variables presented in Table 9 for the back door. In addition to counting passengers from the back door, the second data collector also collected data related to territorial variables identified between stops.

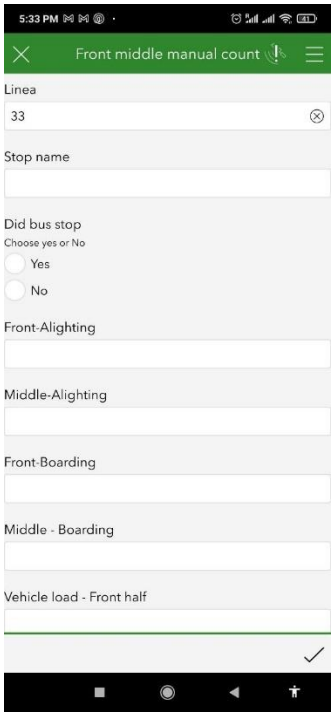


Figure 19. Sample Mobile Survey app

Table 8. Data gathered with 1st survey - Front and Middle door

Fields on app survey	Description	Datatype
Linea	Line number	String
Stop name	Stop name for reference	String
Did bus stop	"Yes" if bus stop in that bus stop, "No" otherwise	String – "Yes", "No"
Front-Alighting	Number of passengers alighting in front door	Int
Middle-Alighting	Number of passengers alighting in middle door	Int
Front-Boarding	Number of passengers boarding in front door	Int
Middle-Boarding	Number of passengers boarding in middle door	Int
Vehicle load – Front half	Total number of on-board passengers from middle section towards front section	Int
Total vehicle load	Total number of passengers on-board	Int
Date and time	Date and time in which the record was taken	Datetime
Stop time	Timestamp in which the record was taken	Int

Table 9. Data gathered with 2nd survey - Back door

Fields on app survey	Description	Datatype
Linea	Line number	String
Stop name	Stop name for reference	String
Did bus stop	"Yes" if bus stop in that bus stop, "No" otherwise	String – "Yes", "No"
Back-Alighting	Number of passengers alighting in back door	Int
Back-Boarding	Number of passengers boarding in back door	Int
Vehicle load – Back half	Total number of on-board passengers from middle section towards back section	Int
Date and time	Date and time in which the record was taken	Datetime M/DD/YYYY HH:MM:SS AM/PM

Stop time	Timestamp in which the record was taken	Int
Traffic	Parameter that represents a high or low traffic identified by collector n. 2.	Int – From 0 to 5. 0 as the lowest traffic level and 5 as the highest traffic level
People flow density	Parameter that represents a high or low people flow density surrounding the bus between stops identified by collector n. 2.	Int – From 0 to 5. 0 as the lowest people flow level and 5 as the highest people flow level
Building density	Parameter that represents a high or low building density surrounding the bus between stops identified by collector n. 2.	Int – From 0 to 5. 0 as the lowest building density level and 5 as the highest building density level

The collected data were automatically sent to a cloud from the mobile phone and, then, exported from ArcGIS Survey123 website. The formats in which data were exported were Shapefile, GeoDatabase and Excel, and included additional ID variables, datetimes of creation and edition and Geo-localisation data, as reported in Table 10

Table 10. ID variables for manual count records

Extra variables	Description	Datatype
ObjectID	Unique Object ID for each record	Int
GlobalID	Unique Global ID for each record	String
CreationDate	Date and time in which the record was sent to the cloud	Datetime – MM/DD/YYYY HH:MM:SS AM/PM
Creator	User that created that record	String
EditDate	Date and time in which the record was edited after being sent to the cloud	Datetime - MM/DD/YYYY HH:MM:SS AM/PM
Editor	User who edited the record	String
x	Longitude in which the record was taken	Float
y	Latitude in which the record was taken	Float

Once all the datasets were downloaded, it was needed to merge them in a single file; this was done manually for each day. Any empty fields were filled in at the end, and each record was validated by re-watching the video recordings and changing the middle door values, if necessary. Additional columns were added such as calculated vehicle load, and total passenger boarding and alighting. Finally, to ensure a good quality data collection, another verification layer was considered. First, two vehicle load -*VL_manual_count* and *VL_calculated*- were

calculated using different variables from the dataset. *VL_manual_count* is related to the number of passengers on board counted by the collectors during the collection, while *VL_calculated* is calculated using the total number of users boarding and alighting on each stop. Then, the variable *VALIDATED?* indicated if both vehicle variables were the same or where different. If they were different, we proceeded to modify the values based on the information collected during the operations time. This allowed us to update any missing passenger.

When this dataset was ready, it was merged with the stop coordinates file provided by GTT because, for further processing, geospatial data are needed. In this file the latitude and longitude are defined for every bus stop in the network. The final manual count dataset is composed of the variables reported in Table 11

Table 11. Column legend for final manual count file

Variable	Description	Datatype
vehicle_or_asset_ID	Unique vehicle ID	String
Day	Day in which data were collected	Datetime -MM/DD/YYYY
Front stops	Stop name reference assigned by the collector n. 1	String
Back stops	Stop name reference assigned during the manual counting by the collector n. 2	String
ID_Fermata	ID of the bus stop assigned by the transport company (GTT)	Int
stop_name	Name of the bus stop assigned by the transport company (GTT)	String
Daytime(GMT)	Day and time in which a specific record was collected. At GMT time zone	Datetime - MM/DD/YYYY HH:MM:SS AM/PM
Daytime(GMT+2or1)	Day and time that in which a specific record was collected. At GMT+2 or +1 time zone	Datetime - MM/DD/YYYY HH:MM:SS AM/PM
timestamp_raw	Epoch timestamp in which a record was collected by collector n. 1. At GMT time zone	Int
timestamp_raw_back_door	Epoch timestamp in which a record was collected by collector n. 2. At GMT time zone	Int
Linea	Line number	String
Corsa	Trip number for a line during the day	Int
Direction	Bus final stop for the trip	String
Numero fermata	Numerical stop order for each ride (corsa)	Int
Fermata effettuata	Indicate if the bus stop is in a specific bus stop	String – Yes/No
Fermata effettuata.1	Indicate if the bus stop is in a specific bus stop	Int – Yes (1) / No (0)
Saliti_front	Number of passengers boarding in front door	Int

Discesi_front	Number of passengers alighting in front door	Int
Saliti_mid	Number of passengers boarding in middle door	Int
Discesi_mid	Number of passengers alighting in middle door	Int
Saliti_back	Number of passengers boarding in back door	Int
Discesi_back	Number of passengers alighting in back door	Int
Boarding_manual	Total number of passengers boarding	Int
Alighting_manual	Total number of passengers alighting	Int
A bordo_f	Passengers on board from the front door to the middle door	Int
A bordo_b	Passengers on board in the back door area	Int
VL_manual_count	Vehicle load calculated by adding up A bordo_f and A bordo_b	Int
VL_calculated	Vehicle load calculated using Boarding_manual and Alighting_manual variables	Int
VALIDATED?	Compare the calculated VL with the VL_manual_count	String - if == → GOOD , else → CHECK
Traffic	Parameter that represents a high or low traffic identified by collector n. 2.	Int – From 0 to 5. 0 as the lowest traffic level and 5 as the highest traffic level
People	Parameter that represents a high or low people flow density surrounding the bus between stops identified by collector n. 2.	Int – From 0 to 5. 0 as the lowest people flow level and 5 as the highest people flow level
Building	Parameter that represents a high or low building density surrounding the bus between stops identified by collector n. 2.	Int – From 0 to 5. 0 as the lowest building density level and 5 as the highest building density level
lat_stop	Latitude of the bus stop	Float
lon_stop	Longitude of the bus stop	Float

3.4.2 Camera-based APC system data

Camera-based APC system is a system already installed and located on the top of each door of the bus. It collects the number of passengers boarding and alighting and estimates the total number of passengers on board after each bus stop. Unfortunately, GTT could only provide data for the month of June 2023 as there were some constraints on accessing the historical data; a total of 14 days of data was ultimately used. The dataset is composed of multiple excel files, per each day and vehicle number, and includes the columns reported in Table 12

Table 12. Column legend for GTT's APC data

Column name	Description	Column datatype
Data e ora	Date and time of the record	DD/MM/YYYY HH:MM:SS
Codice nodo	Stop number in the single record	int
Descrizione nodo	Stop details	string
Codice autista	Driver's code	int
Nome autista	Driver's name	string
Cognome autista	Driver's surname	string
Mezzo	Bus number ID	int
Tipo veicolo	Definition of vehicle type	string
Codice deposito di apparteneza	Not specified	Not specified
Descrizione deposito di appartenenza	Not specified	Not specified
Linea	Bus line in operation	string
Corsa	Not specified	int
Passeggeri presenti	Number of passengers on-board	int
Totale saliti	Total number of boarded passengers	int
Totale scesi	Total number of alighted passengers	int
Saliti porta 1	Number of passengers boarded from door 1	int
Scesi porta 1	Number of passengers alighted from door 1	int
Saliti porta 2	Number of passengers boarded from door 2	int
Scesi porta 2	Number of passengers alighted from door 2	int
Saliti porta 3	Number of passengers boarded from door 3	int
Scesi porta 3	Number of passengers alighted from door 3	int

Saliti porta 4	Number of passengers boarded from door 4	int
Scesi porta 4	Number of passengers alighted from door 4	int
Eccessivo affollamento	Over-crowding	string - "Si", "No"
Tipo di stop	Type of stop	string - "Terminal di arrivo", "Terminal di partenza"

The first step of the pre-processing was related to merging all the files in one. Then, Then, empty values were identified and filled in, and finally, unnecessary columns and rows were removed, as they provided no value for the study.

3.4.3 Wi-Fi-based APC system data

The Wi-Fi-based data are provided by MFA. It collects Wi-Fi probe request packets broadcasted by smartphones in an anonymised manner to ensure privacy of smartphone users. MFA agreed to make these data available to the author for the purpose of this thesis. The data were collected for the same period of the manual count; they are composed by the variables reported in Table 13 and were given as a CSV file with multiple json-type records. Regarding the pre-processing, some data imputation was needed.

Table 13. Column legend for Mobyforall APC data

Column name	Description	Column datatype
timestamp_dt	Date and time of the record	Unix timestamp
vehicle_or_asset_ID	Mobyforall vehicle ID	String
avg_count	Number of passengers counted by the system	Int
date_val	Date of the record	YYYY-MM-DDT00:00:00.000Z
loc_flag	Device's GPS active	string

3.4.4 Environmental data

Various datasets for environmental data were explored. The meteorological and pollution data made available by ARPA were used along with the territorial datasets. The geographical coordinates of the stations from ARPA are given in the Universal Transverse Mercator (UTM) projection, specifically in the WGS84 datum for zone 32N (EPSG:32632). Additionally, the

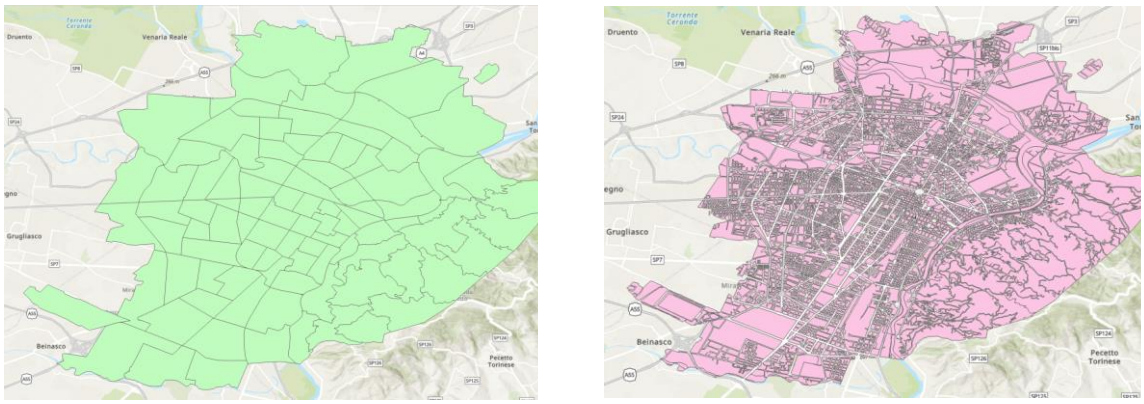
time across the dataset is uniformly expressed in Coordinated Universal Time (UTC). For conversion to local Italian time, one hour (UTC+1) has to be added during standard time (from the end of October to the end of March) and two hours (UTC+2) during daylight saving time (from the end of March to the end of October). Each data point is accompanied by a validation flag that indicates its reliability, as part of the quality management system certified under ISO 9001:2008 by SAI-GLOBAL ITALIA. This flag helps users to assess the trustworthiness of the data.

Territorial vector data and POI

The population distribution by census zones and gender for 2023 in Turin was used as the territorial variable, a sample of this demographic data is reported in Figure 20. Similarly, the vector data collected are related to the territory of Turin. Although multiple shapefiles were available, the ones useful for this study were vector data per census zones and per building, as shown, respectively, in Figure 21a and Figure 21b.

Num_zona	Desc_zona_stat	Femmine	Maschi	Total
01	MUNICIPIO	4855	7624	12479
02	PALAZZO REALE	67	52	119
03	PALAZZO CARIGNANO	2148	1979	4127
04	PIAZZA SAN CARLO - P	2227	2108	4335
05	PIAZZA STATUTO	3889	3672	7561
06	PIAZZA VITTORIO VENE	2455	2056	4511
07	CORSO CAIROLI - PIAZ	2688	2321	5009
08	COMANDI MILITARI - S	2845	2470	5315
09	PIAZZA MADAMA CRIS	5459	5248	10707
09bis	PARCO DEL VALENTIN	12	4	16

Figure 20. Sample of population information by gender in each census zone within the city f Turin as of 2023.



(a) (b)

Figure 21. Territorial vector data for (a) census zones and (b) buildings

The pre-processing methodology for these data is presented in Figure 22. Initially, it was necessary to join the demographic data with the vector census zones. Next, the centroids of the building vectors were calculated. Finally, a geospatial join was performed between these centroids and the census vectors to obtain the population information as centroids attributes for further processing.

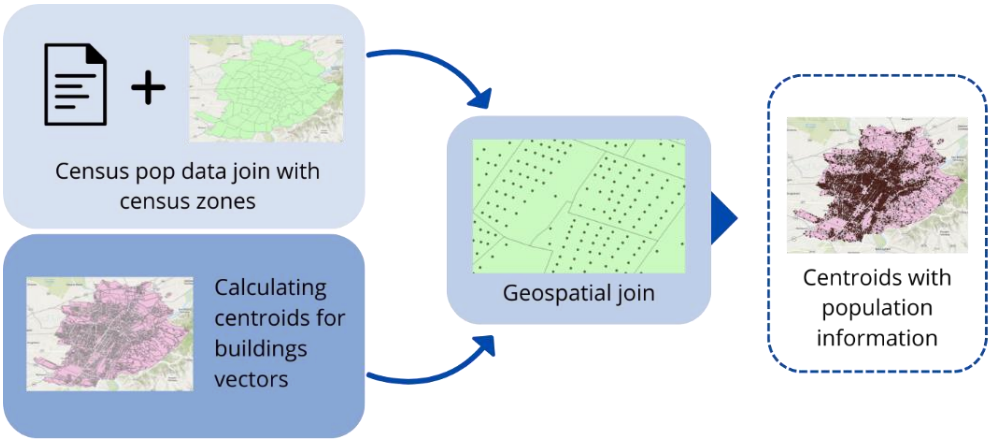


Figure 22. Pre-processing territorial variables

Regarding the POI, two different types of vector data were collected for the north-eastern Italian territory: polygons vectors and points vectors. For the pre-processing, both polygons and vectors were filtered to include only the areas within the scope of the study, i.e. Turin, Moncalieri, Rivoli, Grugliasco, and Collegno. Moreover, as the data contained various POI classes, only the features belonging to relevant classes were kept, excluding those that were less relevant or null, such as traffic lights and traffic cameras (for fines' purposes). The classes considered for this study are reported in Table 14.



Figure 23. POI polygons and points

Table 14. POI classes used for the study

POI CLASSES			
'pharmacy'	'bicycle_rental'	'monument'	'hospital'
'greengrocer'	'bar'	'memorial'	‘swimming_pool'
'bank'	'hotel'	'supermarket'	'wayside_cross'
'atm'	'restaurant'	'cafe'	'food_court'
'butcher'	'beauty_shop'	'travel_agent'	'garden_centre'
'tourist_info'	'pub'	'toy_shop'	'tower'
'florist'	'vending_any'	'clothes'	'water_tower'
'beverages'	'convenience'	'clinic'	'mall'
'computer_shop'	'bicycle_shop'	'shoe_shop'	‘vending_cigarette'
'bakery'	'newsagent'	'hairdresser'	'town_hall'
'fast_food'	'bookshop'	'kindergarten'	'park'
'stationery'	'guesthouse'	'furniture_shop'	'observation_tower'
'playground'	'nightclub'	'optician'	'castle'
'outdoor_shop'	'school'	'theatre'	'stadium'
'dentist'	'car_sharing'	'artwork'	'track'
'toilet'	'post_office'	'fountain'	'golf_course'
'museum'	'chemist'	'car_rental'	'archaeological'
'library'	'police'	'doityourself'	'courthouse'
'doctors'	'pitch'	'shelter'	'college'
'car_dealership'	'veterinary’	'kiosk'	'fire_station'
'laundry'	'attraction'	'picnic_site'	'graveyard'
'hostel'	'recycling'	'car_wash'	'nursing_home'
'sports_centre’	'general'	'dog_park'	'caravan_site'
'market_place'	'wayside_shrine'	'sports_shop'	'viewpoint'
'mobile_phone_shop'	'department_store'	'cinema'	'vending_machine'
'jeweller'	'arts_centre'	'gift_shop'	'ruins'
'community_centre'	'embassy'	'university'	

As shown in Figure 24, to merge the points of interest to our dataset, a spatial join was performed between the POIs and the bus stops location. This involved searching for POIs

within various radius from each bus stop – specifically at distances of 200 meters, 300 meters, and 400 meters – and individual files were created per vector type and distance.

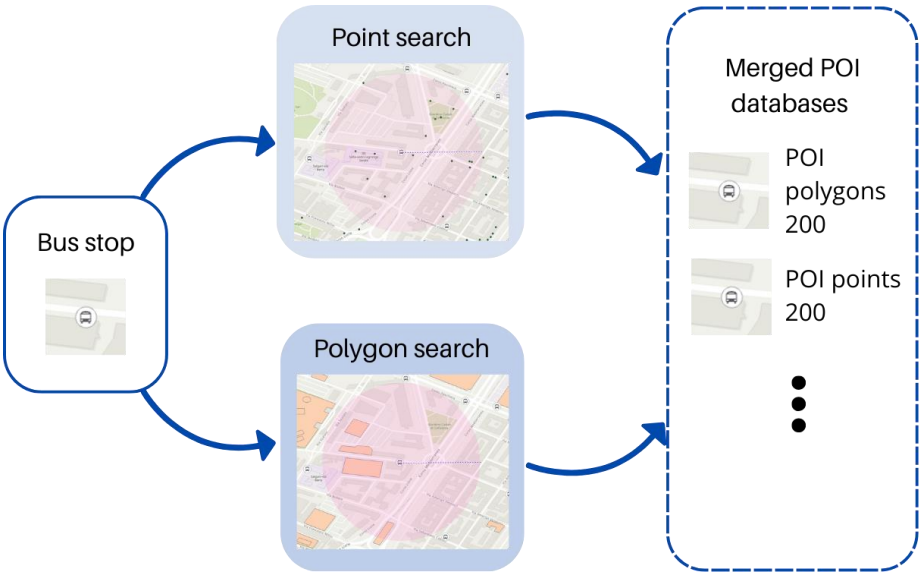


Figure 24. Pre-processing POI

Meteorological data

Meteorological data were provided for each station, variable, and hour throughout every day of 2023. This activity generated multiple files, each containing data for a specific variable at a specific station. Considering the variables that could potentially influence the use of the public transport system, those selected for the study were precipitation, global solar radiation, air temperature, air humidity, and wind speed. Figure 25 shows a sample of one of the many CSV files containing data from ARPA. This specific sample comes from a file containing hourly air humidity percentage data.

data	ora (UTC)	valore	flag validazione
1/1/2023	0:00	99	0
1/1/2023	1:00	99	0
1/1/2023	2:00	99	0
1/1/2023	3:00	99	0
1/1/2023	4:00	99	0
1/1/2023	5:00	99	0
1/1/2023	6:00	99	0
1/1/2023	7:00	99	0
1/1/2023	8:00	99	0
1/1/2023	9:00	97	0
1/1/2023	10:00	95	0
1/1/2023	11:00	98	0
1/1/2023	12:00	97	0

Figure 25 Sample of data for Air humidity from the ARPA dataset

The steps used during the pre-processing are reported in Figure 26. First, the validation flag was checked for all the records. Then, the multiple station files were merged into a file per variable by calculating the average value. Next, the variables were combined into a single file to create a record with columns representing each variable for every hour of the day. Finally, after data imputation, the meteorological dataset was ready for further processing.

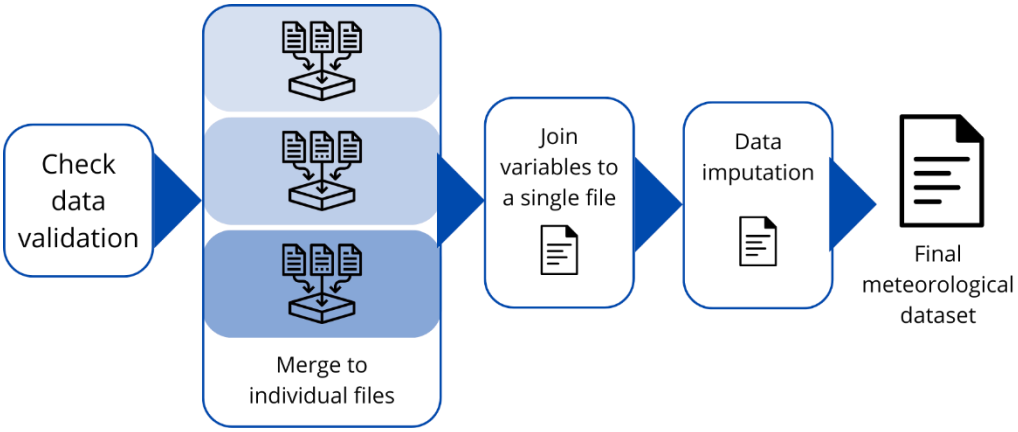


Figure 26. Pre-processing for meteorological data

Pollution data

The pollution dataset is structured similarly to the meteorological dataset, where each station has a file for each variable but on this dataset the data are reported on daily basis instead of hourly basis. Despite multiple parameters available from ARPA, as presented in section 3.2.3, only PM_{10} (Basso volume) is used for this study because it is the most complete dataset considering the multiple sensor stations. An example of the dataset is reported in Figure 27; it shows the concentration of PM_{10} particles measured from the station named "Torino-Grassi".

Data rilevamen	Ora	Id Rete Monitorag	Codice Istat Comune	Progr. Punto Comune	Denomina zione Stazione	Id Paramet	Descr. Parametro	Id Un. misura	Descr. Un. misura	Valore	Stato
1/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	77	V
2/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	56	V
3/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	63	V
4/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	45	V
5/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	51	V
6/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	25	V
7/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	28	V
8/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	29	V
9/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	19	V
10/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	36	V
11/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgramn	73	V
12/1/2023	23:59	13	1272	819	Torino - Gra	PM10_GBV	PM10 - Basso Volume	23	microgrammi / metro cu		E

Figure 27 Example of PM10 data from Torino - Grassi station

To identify the PM_{10} values in the territory, an inverse distance weighting model (IDW) was used to apply spatial interpolation for each day of the data collection. Then, each daily raster model values were extracted to the building vectorial centroids in order to obtain the daily values in a single centroids file. This pre-processing methodology is reported in Figure 28

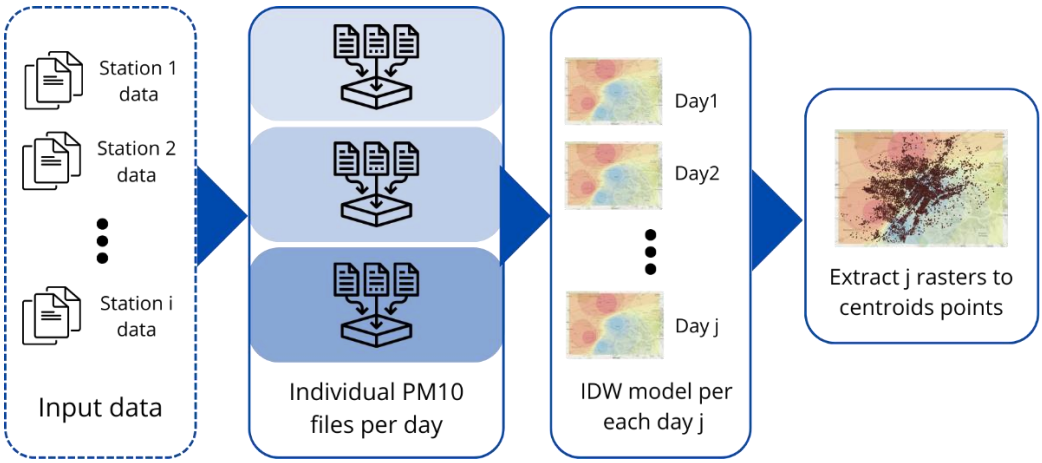


Figure 28. Pre-processing for pollution data

3.5 Data processing

In this step, we focus on merging all the datasets in a single one. This involves creating a unified database that integrates trip information between stops with available territorial variables, serving as the foundation for our analysis. The data files to merge were CSV and ShapeFiles (SHP); thus, to facilitate the analysis of spatial data, it was imperative to convert the collected data at some point of the process (initially in Dataframe format) into GeoDataframe format. The conversion was executed using Python, leveraging the capabilities of the Pandas and GeoPandas libraries. The process is outlined in several steps, as reported in Figure 29 and explained in the following sub-sections.

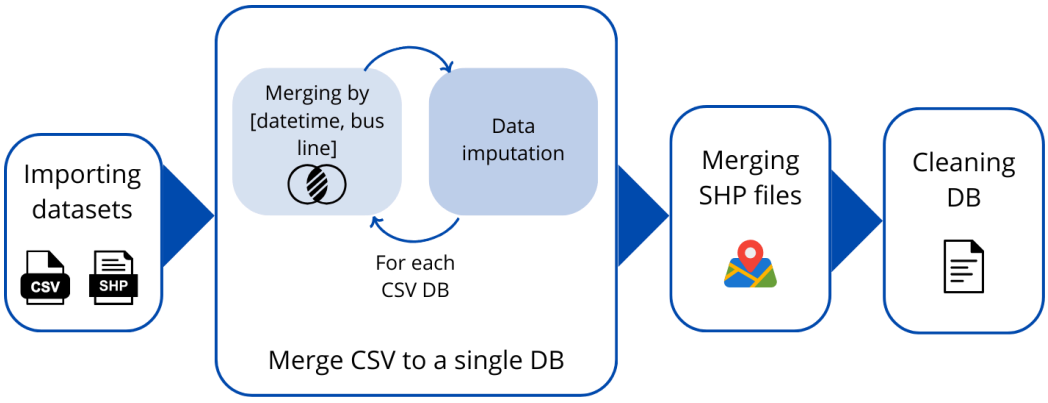


Figure 29. Processing methodology

3.5.1 Importing datasets

From the pre-processing two different types of datafiles were obtained: CSV and SHP and each of them was treated differently. The CSV files available were manual count, camera APC, Wi-Fi APC and meteorological data, while the SHP were territorial data, POIs and pollution data.

The CSV files containing the datasets were loaded into a Pandas DataFrame (DF). This structure facilitated the manipulation of the data, given its inherent capabilities for handling tabular data efficiently. The dataset was assumed to include columns representing latitude and longitude coordinates of the specific bus stop for each record, essential for the geographical representation of the data.

On the other hand, the SHP files were imported as GeoDataFrame (GeoDF) using GeoPandas library and their coordinate system were then converted to EPSG:4326 to ensure homogeneity. No additional processing was required for the territorial data and POIs after being imported. However, the pollution dataset was structured with one column for each date (header) and PM_{10} values as rows for each centroid location. The date and PM_{10} values needed to be saved as separate columns, so the dates and PM_{10} values for different days were consolidated into two single columns for easier matching and merging. This operation was carried out using the Pandas *melt* function

3.5.2 Merging the CSV Files

The CSV files available were merged by day and time. The merging was performed using a Pandas' function called *merge_asof* which merges by key distance, and the assigned keys for these datasets were the datetime variable (i.e. the variable that represented the day and time that a specific record was collected). The main reference dataset was the manual count, meaning that the merge was primarily made between this dataset – left DF – and the others – right DF. Merging was conducted with a "nearest" search, which selects the row in the right DF whose 'on' key is closest to the left key within a specific time tolerance. In summary, the merge was done considering the nearest timestamp and the stop number.

From the manual count, the average and maximum time distance between the stops were equal, respectively, to 32 and 62 seconds, hence initially the time tolerance was set to one minute. However, after evaluating the number of NaN values (see Table 15), there was a significant loss of data. Thus, different time tolerances were tested, and it was decided to use a two-minute tolerance as a good compromise with the quantity of data. After analysing the data, the missing values are presumably related to the long waiting time at bus terminus, or last stop. During the manual count, there were days when waiting times at terminus reached up to 10 minutes before starting the trip in the other direction.

Table 15. NaN values count, after merge, per time tolerance

Dataset	NaN values – 1 min	NaN values – 2 min
camera APC	313	274
Wifi APC	173	169
Meteorological data	0	0

Once a dataset was merged, data imputation was performed before proceeding to merge the next dataset. This step ensured correct imputation, considering the different data behaviours

and ensuring data quality after combining the datasets. The NaN values from vehicle load, boarding and alighting of any APC system were filled with 0, as a penalty, to consider the potential data missing during real operations.

3.5.3 Merging SHP files

The SHP files were merged with the bus location coordinates from the main DB using a spatial join. As a first step, the main DF was converted into a GeoDF by applying a function that creates Point geometries of the bus stops from the longitude and latitude columns. Then the spatial join (*sjoin_nearest*) feature of GeoPandas was used to merge the two GeoDF – the value of the point nearest to the bus stop is chosen and assigned to the record. Next, when needed, a date filter was applied to align the values with only the dates present in the main GeoDF records. Since the spatial join can list multiple points nearest to a stop, a duplicate removal step was implemented to keep only the closest point data for each stop for each day. This method was consistently used for all SHP files.

3.5.4 Dataset segregation

In this step the final modifications to the DB were done before proceeding with the data fusion.

- individual files of the final DB were made for each POI distance, meaning that at the end three final DBs were obtained, with different POI values;
- to facilitate temporal analyses, datetime columns were converted to Unix timestamps;
- redundant or irrelevant columns were removed to streamline the dataset and focus on the most pertinent features. The final features for the main DB are reported in Table 16;
- the main DB was split into training and testing sets with a 70:30 ratio. Exogenous variables for both train and test sets are prepared by dropping the target column. The selected target variable was Vehicle load, or *VL_MC*

Table 16. All features of main DB

Variable	Description
vehicle_or_asset_ID	Unique vehicle ID
ID_Fermata_x	ID of the bus stop assigned by the transport company (GTT)
date_timestamp_UTC+_ MC_unix	Day and time in which a specific record was collected. At GMT+2 or +1 time zone
Linea	Line number
Corsa	Trip number for a line during the day
Direction	Bus final stop for the trip
Numero fermata_x	Numerical stop order for each ride (corsa)
Fermata effettuata_x	Indicate if the bus stop in a specific bus stop
Boarding_front_MC	Number of passengers boarding in front door from manual counting
Alighting_front_MC	Number of passengers alighting in front door for manual counting

Boarding_mid_MC	Number of passengers boarding in middle door for manual counting
Alighting_mid_MC	Number of passengers alighting in middle door for manual counting
Boarding_back_MC	Number of passengers boarding in back door for manual counting
Alighting_back_MC	Number of passengers alighting in back door for manual counting
Total_boarding_MC	Total number of passengers boarding for manual counting
Total_alighting_MC	Total number of passengers alighting for manual counting
VL_MC	Vehicle load from manual counting
Traffic	Parameter that represents a high or low traffic identified by collector n. 2.
People	Parameter that represents a high or low people flow density surrounding the bus between stops identified by collector n. 2.
Building	Parameter that represents a high or low building density surrounding the bus between stops identified by collector n. 2.
hour	Hour in which the record was collected
VL_GTT	Vehicle load estimation provided by the commercial APC system from GTT
Total_boarding_GTT	Total number of passengers boarding for GTT
Total_alighting_GTT	Total number of passengers alighting for GTT
Boarding_front_GTT	Number of passengers boarding in front door for GTT
Alighting_front_GTT	Number of passengers alighting in front door for GTT
Boarding_mid_GTT	Number of passengers boarding in middle door for GTT
Alighting_mid_GTT	Number of passengers alighting in middle door for GTT
Boarding_back_GTT	Number of passengers boarding in back door for GTT
Alighting_back_GTT	Number of passengers alighting in back door for GTT
Next stop ID	ID of the bus stop next to the current stop
WiFi_predicted_count	Vehicle load estimation by a very low-cost Wi-Fi based APC
value_pollution_PM10	PM_{10} value
valore_temp_aria	Air temperature value
valore_prec	Precipitation value
valore_hum	Air humidity value
valore_radi	Solar radiation value
valore_wind_speed	Wind speed value
num_stat_zone_popu	Census zone number ID
Total_popullation	Total population in the area
POI VARIABLES - 107 COLUMNS OF POI VARIABLES	
Join_Count_400mt	Total POI found in 400mt surroundings
day_of_week	Day of the week in which the record was taken
day_num	Day of the month in which the record was taken
num_month	Number of the month in which the record was taken

3.5.5 Encoding and Normalisation

The dataset consisted of a mix of numerical and nominal categorical variables. Categorical columns, i.e. *vehicle_or_asset_ID*, *Direction*, *day_of_week*, *ID_Fermata_x*, *Numero fermata_x* and *Next stop ID*, were encoded. There were two different approaches initially considered for encoding the categorical variables:

- **LabelEncoder**: label encoding assigns a unique integer to each category. While simple to implement, it introduces an ordinal relationship between categories, which may not exist in the data. This can mislead models into interpreting the encoded values as ordinal features;
- **TargetEncoder**: on the other hand, by considering the target variable during encoding, it helps in capturing more meaningful insights from the categorical variable, which can be especially beneficial when the relationship between the category and the target is non-linear or more complex.

At the end, *target encoding* was finally used. This approach involves replacing each categorical value with the mean of the target variable corresponding to that value. This encoding technique is particularly useful in reducing the dimensionality of the data while preserving the predictive power of categorical features. By incorporating information from the target variable into the features, there is a risk of data leakage, especially if the data are not properly split between training and testing sets before applying target encoding. To overcome such issues, testing of the training split is made before performing target encoding and target encoding model is fit according to the training data and then the test dataset is transformed using this fit to avoid data leakage.

Once encoding was completed, normalisation was applied using *MinMaxScaler*. *Normalization* ensures that all features are on a comparable scale, which is crucial for the performance of machine learning algorithms. By scaling the features to a range between 0 and 1, we mitigate the risk of features with larger scales dominating the model training process.

3.5.6 Resampling and Imputation

Before applying algorithms on this dataset, some preparatory steps were needed to be done on the dataset. Firstly, the original dataset was loaded from an Excel file. The timestamp column, *date_timestamp_UTC+_MC*, was then converted to a datetime format, ensuring that all timestamps were not time zone aware. This conversion was essential for consistent time-series analysis. The dataset was then sorted by the timestamp to maintain the chronological order of observations.

Next, to handle potential large gaps in the data, the data were resampled within each group of *Day*, *Linea*, and *Corsa*. The data were resampled to 30-second intervals, using forward fill

to handle missing values within a permissible gap of 15 minutes. If the gaps exceeded this threshold, the corresponding entries were marked as NaN and subsequently filled using forward fill followed by backward fill to ensure continuity.

But, in the above way of resampling, there were still significant gaps in time in the dataset because the data collection process had to adapt to the availability of the bus, so for one day there might be data from 09:00 AM to 02:00 PM, and for another day from 11:00 AM to 04:00 PM. This makes it difficult, especially for models like SARIMAX, to learn any seasonality. To investigate if there was any way to overcome this limitation, some imputation algorithms were attempted so that all days would have data for the same hours, allowing for the learning of any seasonal patterns. The chosen imputation mechanism was K-Nearest Neighbour (KNN) Imputation. KNN imputation fills missing values by averaging or voting based on the 'k' nearest neighbours' values. This method maintains the relationships and patterns within the dataset, often leading to more accurate imputations than simple mean or median filling.

There was a concern that using the KNN imputation approach on the full dataset might compromise the quality of the validation strategy. Therefore, another approach was attempted, involving K-Nearest Neighbour (KNN) imputation solely on the training dataset. This method ensures that the imputed values do not introduce bias during model validation, preserving the integrity of the process. By applying KNN imputation only to the training data, the model learns from a more consistent dataset, while the test data remains untouched. This ensures that the evaluation reflects the model's performance on genuinely unseen data, enhancing the reliability and validity of the performance metrics and providing a more accurate representation of its real-world applicability. The application of KNN imputation facilitated a more uniform dataset, which is critical for models sensitive to temporal patterns, such as SARIMAX. By ensuring consistent hourly data across all days, the model could better detect and learn underlying seasonal trends. This adjustment aimed to enhance model performance and provided a more robust framework for future data collection and analysis, potentially leading to more accurate and reliable forecasting outcomes. Feature selection was performed only using the non-resampled dataset to rely fully on real values and associations. However, the data fusion models were applied to all three datasets for comparison: the dataset without KNN imputation, the dataset with KNN imputation for the full dataset, and the dataset with KNN imputation only for the training dataset.

3.6 Data fusion models and validation

Data fusion models integrate information from multiple sources to create a comprehensive and coherent representation of the data. By combining diverse datasets, these models enhance the accuracy and robustness of predictions and insights. The primary advantage of data fusion lies in its ability to leverage complementary information, thereby mitigating the limitations of individual data sources, and this approach is particularly valuable in complex domains where single-source data may be insufficient or biased. For this study, the purpose of data fusion is to estimate the vehicle load and to achieve this; thus, a methodology for testing and comparing multiple data fusion models is proposed. Six different algorithms were used: three statistical

models – Bayesian model, SARIMAX model and FB Prophet model – and 3 ML/DL models – XGBoost, LightGBM and LSTM model (Deisenroth *et al.*, 2020; Korstanje, 2021).

A learning model should be capable of generalising the "knowledge" learned from training data to estimate unseen instances. Thus, recording a small prediction error and no-overfitting would be one of the main goals. However, the estimation of errors requires a ground-truth parameter to compare it with the predicted values. For the purpose of this study, the ground-truth is the survey data. Thus, the features used for estimating the fused value were define on section 3.6.2. Each model required tuning to define the hyperparameters and ensure a good prediction performance. Then, as reported in Figure 30, the vehicle load was predicted using the testing data, and the results were eventually compared with the ground truth during validation step. In addition to numerical evaluations, visual comparisons were made through scatter and line plots. These plots compared actual versus predicted values for both validation and test sets, providing a visual insight into the models' predictive performance.

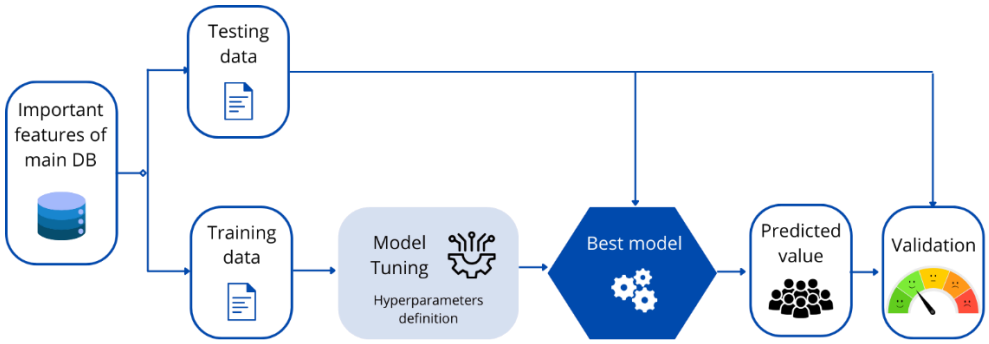


Figure 30. Data fusion and validation methodology

3.6.1 Validation metrics

After the predictions were made on the test data, these were evaluated using metrics such as SMAPE, Mean Absolute Error (MAE), and Pearson and Spearman correlation coefficients. These metrics provide a comprehensive assessment of model performance, capturing both accuracy and the strength of linear and rank correlations. Pearson and Spearman correlations provide insights into different aspects of the relationship between predicted and actual values. With Pearson correlation the strength and direction of linear relationship can be measured while with Spearman correlation the monotonic relationship can be identified, capturing both linear and non-linear trends (Willmott and Matsuura, 2005; Hauke and Kossowski, 2011; Chicco *et al.*, 2021). SMAPE formula is reported in Equation (9) and (10); MAE is given by Equation (12) and corresponds to the average absolute differences between the predictions and the ground truth; lower the MAE, better the prediction:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (12)$$

Where n is the number of records, y_i is the ground truth for the record *i* and $\hat{y}_i$ correspond to the predicted value for the record *i*. Pearson and Spearman correlation are defined, respectively, by Equations (13) and (14).

$$p = \frac{cov(y_i, \hat{y}_i)}{std(y_i)std(\hat{y}_i)} \quad (13)$$

Where $cov(y_i, \hat{y}_i)$ is the covariation function between the ground truth and the predicted value, $std(y_i)$ is the standard deviation of the ground truth and $std(\hat{y}_i)$ is the standard deviation of the predicted value.

$$r = \frac{cov(ry_i, r\hat{y}_i)}{std(ry_i)std(\widehat{ry}_i)} \quad (14)$$

Where $cov(ry_i, r\hat{y}_i)$ is the covariation function between the rank of ground truth and the rank of predicted value, $std(ry_i)$ is the standard deviation of the rank of ground truth and $std(r\hat{y}_i)$ is the standard deviation of the rank of predicted value.

3.6.2 Feature Selection

Feature selection is very important in model development as it helps to build better classifiers, remove irrelevant and redundant data, reduce computation time, improve learning accuracy, and facilitate better understanding for the learning model or data. (Cai *et al.*, 2018; Khaire and Dhanalakshmi, 2022). Feature selection involves calculating a score for each feature based on a specific criterion, with higher values indicating better features. These criteria vary by feature selection method, which are classified into three categories: (i) filter: features are selected independently of the learning algorithm based on statistical measures, requiring less computation time; (ii) wrapper: these methods are computationally more expensive as they train the algorithm using subsets of features iteratively, selecting the best subset based on classifier results; and (iii) embedded methods: these use ensemble and hybrid learning methods for feature selection. Due to their collective decision-making process, embedded methods often outperform the other two models (Venkatesh and Anuradha, 2019; Kavya D, 2023).

Given the vast number of potential features available in the main DB, the methodology in this study involved calculating feature importance scores by testing different feature selection algorithms. These scores were then combined and visualised to identify the most significant features. This multi-faceted approach to feature selection helps ensure that the most relevant features are retained for model training.

The methods used in this study included mutual information regression via *SelectKBest*, a filter-based method that selects the top *k* features with the highest mutual information scores relative to the target variable, and three embedded methods: Random Forest, LightGBM, and XGBoost. In each case, a higher score indicates greater dependency or association on the target variable or the higher influence in reducing the loss function for the embedded methods, highlighting the most important features for target variable estimation. This study identified the best features for two scenarios; first, with the main DB encoded and normalised, and second, with the main DB encoded but not normalised. From this analyses, the top 20 features were selected for further processing.

3.6.3 Statistical Models

In the context of data fusion, statistical models can synthesise information from diverse sources to derive meaningful insights. Based on the literature, the three models presented in this study are Bayesian Ridge regression, SARIMAX, and FB Prophet, as these provide a robust framework for data fusion in the literature, enhancing predictive accuracy across various fields by integrating multiple data sources or identifying underlying patterns in time series data. Bayesian regressions are effective in data fusion due to their ability to incorporate prior knowledge and iteratively update predictions as new data becomes available. This adaptability is invaluable when integrating data from multiple, potentially disparate sources. SARIMAX model, or Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors, excels in time series forecasting by accounting for seasonal patterns and incorporating external variables, which is essential when merging time-dependent data from various origins. On the other hand, FB Prophet, developed by Facebook, is designed for scalability and ease of use, making it a powerful tool in data fusion tasks with complex datasets. It handles missing data and outliers effectively and allows the inclusion of holidays and seasonality effects, thus providing robust forecasts. By leveraging these models, data fusion efforts can achieve higher accuracy and reliability in their predictions and analyses.

Model tuning was made for each of the selected models. All the models were first fit and trained using the training dataset and then evaluated for their performance on the testing dataset, however their tuning technique differs among them. The implementation of Bayesian Regression was carried out using the *BayesianRidge* class from the scikit-learn library. It has four hyperparameters—alpha_1, alpha_2, lambda_1, and lambda_2— that control the shape of the priors on the model coefficients and the noise variance. Specifically, alpha_1 and alpha_2 are the hyperparameters for the precision of the Gaussian priors on the weights, while lambda_1 and lambda_2 are the hyperparameters for the precision of the prior on the noise. To identify the best combination of these parameters, a hyperparameter tuning process was conducted using K-Fold cross-validation with five splits. The hyperparameters evaluated were:

- alpha_1: [1e-6, 1e-3, 1]
- alpha_2: [1e-6, 1e-3, 1]
- lambda_1: [1e-6, 1e-3, 1]

- lambda_2: [1e-6, 1e-3, 1].

The cross-validation process identified the optimal hyperparameters that minimised the validation mean squared error (MSE). The final Bayesian Regression model, trained with these optimal hyperparameters, was evaluated on the test set. Model performance was then assessed using metrics presented in section 3.6.1.

For SARIMAX model, the pmdarima library's *auto_arima* function was used. This method involves a comprehensive search over specified ranges of non-seasonal (p, d, q) and seasonal (P, D, Q, m) parameters, evaluating models using Akaike Information Criterion (AIC) to balance goodness-of-fit and simplicity. AIC is a measure that balances model fit and complexity, penalising models with more parameters to avoid overfitting. The model with the lowest AIC value was selected. '*auto_arima'* was applied to the training data to determine the best SARIMAX parameters and the best SARIMAX model was selected based on the lowest AIC value. The parameters evaluated were:

- **Non-seasonal parameters:**
    - p**:** Number of lag observations (AR part).
    - d**:** Number of differencing steps (I).
    - q**:** Size of the MA window.
- **Seasonal parameters:**
    - P**:** Seasonal AR terms.
    - D**:** Seasonal differencing steps (I).
    - Q**:** Seasonal MA terms.
    - s**:** Time steps in one seasonal period.

For the prophet model, The Prophet library, developed by Facebook, was utilised to build the model and a looping structure was used for hyperparameter tuning to enhance the performance of the time series model. The aim was to systematically evaluate different parameter combinations to identify the optimal configuration for the model.

A range of values for each parameter was defined to explore various combinations. Each combination of parameters was used to configure and train the Prophet model on the training data. The model's performance was then assessed on the test data. The evaluation involved calculating accuracy metrics such as percentage accuracy, Pearson correlation, and Spearman correlation to determine the effectiveness of each parameter combination.

Parameters and Values

- **Period**: it defines the period of the added seasonality component. The tested values were:
    - 1/24 (hourly seasonality)

    - 1/48 (half-hourly seasonality)
    - 1/96 (quarter-hourly seasonality)

- **Daily Seasonality**: it indicates whether to include daily seasonality. The tested values were:

    - True
    - False.

- **Fourier Order**: it controls the number of Fourier terms for seasonal components. The tested values were:

    - 3, 6, 9, 12, 15, 18, 21, 24.

- **Seasonality Mode**: it specifies the mode of seasonality. The tested values were:

    - additive
    - multiplicative.

3.6.4 Machine and Deep Learning Model

Two boosting algorithms and one RNN were evaluated on our database: (i) *XGBoost* is a tree-based ensemble method using gradient boosting. It well handles sparse data, includes L1 and L2 regularisation to prevent overfitting, and it is highly scalable, making it effective for large datasets; (ii) *LightGBM* is similar to XGBoost but optimised for speed and memory usage. It uses a histogram-based algorithm for faster training and better efficiency, supports large datasets with distributed computing, and can be trained with reduced dataset size without significant accuracy loss; and (iii) *LSTM* which is a recurrent neural network designed for sequence data. It processes sequences, learns long-term dependencies due to its gated architecture, and is flexible for various sequence lengths (Chen and Guestrin, 2016; Ke *et al.*, 2017b; Che, 2021; Chaturvedi *et al.*, 2022).

XGBoost, LightGBM and LSTM models were trained using a processed and feature-selected dataset. From the training set with 70% of the data from the data preparation stage, 10% was further sub-divided and used for validation, which equates to 7% of the full dataset. This approach ensures that the model has a substantial amount of data for learning while maintaining a dedicated subset for tuning and another for unbiased evaluation.

Allocating 70% for training allows the model to capture the underlying patterns effectively, while using 10% of this training set for validation helps in fine-tuning the model's hyperparameters, reducing the risk of overfitting. The remaining 30% reserved for testing offers a reliable measure of the model's generalisation capabilities on unseen data. This balanced data splitting strategy maximises the use of available data, ensuring robust model training and comprehensive performance evaluation. This balanced data splitting strategy maximises the use of available data, ensuring robust model training and comprehensive performance evaluation, and it also ensures that all models, including the statistical models

from the previous section, are evaluated on the same test dataset, assuring a valid and reliable comparison.

Before proceeding with training the model for the prediction, the model tuning was done using Gridsearch. *Gridsearch* is a widely used technique in hyperparameter tuning for machine learning algorithms, offering a systematic approach to test a predefined set of hyperparameters to find the optimal combination for a model. It is especially popular for its simplicity and comprehensiveness, albeit sometimes limited by high computational demands in complex models (Amirabadi *et al.*, 2020). This technique facilitated the selection of the optimal model, which was then used to derive the final fused value from the test dataset, prior to applying the performance metrics. During the Gridsearch, each model, although based on the same algorithm, was tested with a range of hyperparameters, utilising the validation subset of the training dataset for validation purposes. The results are then analysed one model at a time to understand possible areas of improvement and to make future improvements to the accuracy of the models under study.

For XGBoost and LightGBM models, *RandomizedSearchCV* was employed to optimise the parameters n_estimators, max_depth, and learning_rate. This approach involved randomly sampling from specified ranges to identify the best combination of hyperparameters. The ranges tested were:

- n_estimators – between 10 and 1000, with a total of 10 samples;
- max_depth – between 10 to 110, with a total of 11 samples;
- learning_rate – 0.01, 0.05, 0.1, 0.2.

This methodology allowed for a comprehensive search across a broad range of values, improving the likelihood of finding an optimal set of parameters that enhance model performance. Then, training was performed and predictions were generated for both the validation and test sets. The models' performance was evaluated following the validation metrics explained in section 3.6.1.

The hyperparameter tuning for the Long Short-Term Memory (LSTM) model was conducted to identify the optimal configuration for maximising model performance. This methodology employed the *Hyperband* tuner from the keras_tuner library to systematically evaluate various parameter combinations. The hyperparameters tested included:

- Units: 50, 100, 150;
- Dropout Rate: 0.2, 0.3, 0.4;
- Optimizer: 'adam', 'rmsprop';
- Loss Function: 'mean_squared_error', 'mean_absolute_error'.

To prepare the data for the LSTM model, sequences were created to transform the dataset into input-output pairs with a sequence length of 60. This ensured that the data were appropriately structured for time series forecasting. Then, for model tuning, a model-building function was defined with specified hyperparameters, and the *Hyperband* tuner was set to

minimise validation loss. Parameters such as `max_epochs` and `hyperband_iterations` were specified to control the tuning process, and an early stopping callback was implemented to halt training if validation loss did not improve for five consecutive epochs. The `tuner.search` method systematically evaluated different combinations of hyperparameters on the training data.

The best hyperparameters identified by the tuner were used to build and train the final LSTM model, which was then evaluated on the test set. Predictions were compared with actual values using accuracy and correlation metrics, providing an assessment of the model's effectiveness. This systematic approach ensured the identification of an optimal hyperparameter set, enhancing the LSTM model's performance in time series forecasting.

3.7 Comparative Analysis

The assessment of AI models versus traditional statistical models encompasses a range of factors, such as performance, scalability and computational demand. Understanding the impact of each of these models on the criteria is essential in this section of the study. For this reason, a Decision Matrix, or Pugh matrix analysis, is implemented, which is the simplest form of Multiple Criteria Decision Analysis (MCDA). It is a structured method used to evaluate and prioritise a list of options based on multiple criteria, helping decision-makers systematically compare different alternatives by scoring them against a set of predefined factors, weighted according to their relative importance This comprehensive comparison aims to highlight the distinct characteristics and suitability of each model type across various computational and analytical requirements, guiding optimal model selection based on the specific needs of the task at hand (Paradkar, 2016; Tarantino, 2022; Mind Tools, 2024). For this reason, in this step, not only the models were analysed, but a replicable tool was also provided to determine which model is better suited for a specific use case based on criteria derived from the results of this study.

The decision matrix provides a clear, objective framework for decision-making, ensuring that all relevant factors are considered and appropriately weighted. First, the factors were normalised using MinMax normalisation, and factors where a lower value indicates a better performance were inverted. Then, each option is evaluated by assigning values to each factor, and these values are then multiplied by the corresponding weights. The weighted values are summed to produce an overall score for each option, enabling an easy comparison of alternatives. The model that had provided the best performance in the analysis is the one that had the higher score, and the score of the model is calculated using Equation (15), in which m is the model to evaluate, $V_{(m,f)}$ is the value of the model in m for the factor f and W_f correspond to the weight assigned to the factor f

$$s_m = \sum_{f=1}^{F} V_{(m,f)} \times W_f \quad (15)$$

3.7.1 Factors definition and calculation

The examination extends to model complexity and interpretability, contrasting the extensive parameterisation in ML models against the relative simplicity of statistical models, and assessing the ease with which a model's decision-making processes can be comprehended. This comprehensive comparison aims to highlight the distinct characteristics and suitability of each model type across various computational and analytical requirements, guiding optimal model selection based on the specific needs of the task at hand. For this reason, the main factors considered for the analysis are related to training time, resource usage, accuracy, error and correlation. These were calculated during the data fusion step and were chosen based on the most common constraints faced during building a model and the expectations from its implementation. Usually, the final user of the model expects very high accuracy without considering factors such as data quantity and resource usage. So, by including diverse performance factors, it could be easier for the final user to decide which model is best for their application. The factors were calculated using a learning curve with *y* as performance and *x* as simple size, starting the training with low number of records; then is increased and the performance is evaluated to understand if there is any improvement. The factors considered for this curve were r^2 score, average memory usage and average time usage. The full data imputation dataset was used for evaluating this step:

1. average time usage: it measures the training time for each model during the training phase. It was calculated using different python functions that measured the time before and after the training process. The value was determined using a learning curve, with the final variable representing the average training time across different sample sizes;

2. average Resource Utilisation: it measures the memory usage during model training. Similar to average time usage, it was obtained using a learning curve and represents the average memory usage estimated across different sample sizes;

3. data quantity: it measures the good data quantity needed to get a good performance. It was evaluated using the r^2 score and implemented using scikit-learn library, which typically returns the r^2 score for regression models. For models in which this function was not available, other methods were implemented. A value was then assigned to describe the data quantity required for good model performance, ranging from 1 (to describe small dataset is sufficient), to (to describe a larger dataset is needed);

4. accuracy, error and correlation were calculated for the test dataset using the methods explained in section 3.6.1

3.7.2 Weights definition

The weights could be potentially assigned empirically based on the experience of the decision-maker. However, for this study a more quantitative approach was used. The weights were defined based on the Paired Comparison Analysis, which is a decision-making tool used to evaluate a set of options by comparing them in pairs to establish preferences. This method helps prioritise options by considering them two at a time, making it easier to manage complex decisions involving multiple criteria (Farr and Faber, 2023). The steps of this method were first to list the factors and assign a letter to each of them (A, B, C, D, etc.). Then, they were placed in a comparison matrix with both the rows and columns representing the lettered factors. For each blank cell, the row feature and the column feature were compared to determine which option was more important, and the letter of the more important feature was written in the cell. The difference in importance between the features was scored on a scale from zero (no difference/same importance) to three (major difference/one much more important than the other). Finally, the results were consolidated by summing the values for each option and converting these totals into percentages of the overall score. These percentages were used as the weights for the decision matrix.

Chapter 4

Results

This chapter presents the results of the thesis that have been achieved so far. Firstly, a review of the APC systems in the market is presented, then the results from the pilot study using the low-cost system are shown and discussed in section 4.1. Then, the model performance evaluation results are presented and finally the decision matrix results are reported.

4.1 Pilot study

The development of the low-cost camera-based APC system allowed understanding some insights regarding the technical functionality in both controlled and real-application experiment. Moreover, some errors on data collection were identified while doing the on-field application. In the following sections the results for developing the low-cost APC system are presented.

4.1.1 Model implementation in controlled environment

All the results presented in the following section are related to the controlled environment experiment. In particular, data collection and simulation are described, followed by the choice and the implementation of the required algorithms to perform a proper counting.

- Model implementation

Due to the computational power required for both training and testing, the first choice to make is related to the environment in which running the detection algorithm. According to the need of processing large amounts of images, Google Colab was the suitable solution for its powerful GPU: this allows the dataset to be uploaded on Google servers, speeding up the overall processing. The choice of using Google Colab is also justified by the supported programming language, which is Python, that offers great benefits: among them there is the access to a large number of libraries and frameworks for machine learning as well as a wide community and documentation to look up.

The first decision is related to the object detection algorithm. Once the data was collected on the controlled environment, this dataset was employed as input for training the algorithm. For this, a pre-processing stage is needed to extract the frames that compose the input dataset. A useful tool for performing this step is OpenCV (Computer Vision) Python library. Then, the training set of frames were labelled: a .txt file containing the bounding boxes coordinates for each target person is associated to each photo. This task was fulfilled using the MakeSense.ai software.

For any YOLO algorithm, to start the training step, two files were needed: a customized .yaml file containing the classes name and the corresponding identifying number, and a .pt file containing the initial weights of the neural network. This later defines also the network structure, allowing different configurations that depend on the density of the used neurons (s, m, l, x). The output of this process is the final best weights that would be used to perform detection and tracking on the test video.

The goodness of these weights, and so the corresponding choice of the version to use, is evaluated through accuracy, recall and metrics. After having trained different versions of YOLO, we chose YOLOv5m (Hou *et al.*, 2019; Wu *et al.*, 2021; Rahaman *et al.*, 2022), which provided the best results, with Precision = 0.95, Recall = 0.89, mAP@.5 = 0.96, mAP@[.5:-0.95] = 0.62. The next step is related to the tracking and counting stage, for which DeepSort algorithm has been considered as the best option.

For both tracking and counting algorithm, it is not needed the training phase because the algorithm receives the pre-trained weights of the previous step as the input. Together with the test video, the counting results will be obtained. The goodness of the tracking phase strongly depends on the confidence level reached during the detection of each subject, avoiding ID losses and re-associations to the same person already detected in the previous frame. This procedure leads to the actual counting implementation which consists of drawing two regions of interest over the frame: the choice of the ROI width and location depends on the area of the frame where the tracking results to be more effective.

The obtained results are shown in Table 17, using YOLOv5m as detector, DeepSort as tracking algorithm and two rectangular ROI placed one above the other for the counting. Notice that the reported results are obtained by considering the number of True Positives (TP), False Positives (FP) and False Negatives (FN) both for boarding (B) and alighting(A) people over 10 different videos. Also, the true number of alighted and boarded passengers is highlighted (Total_A/Total_B).

Table 17. Controlled environment results

TP_alighting	FP_alighting	FN_alighting	Total_A	TP_Boarding	FP_boarding	FN_Boarding	Total_B
15	3	4	19	19	1	3	22

Table 18. Performance indicators for the controlled experiment

	Accuracy	Precision	Recall
Boarding	0.826	0.95	0.864
Alighting	0.68	0.833	0.789

From the controlled environment the results show that boarding has a better performance than alighting. Especially regarding accuracy, where boarding got a detection accuracy of 82.6% while alighting got 68%.

4.1.2 Algorithm refinement

In this section the results related to data collection the algorithm refinement. During the data collection on-field, different parts of the system are tested to check their correct working in a real-life scenario as well as the effectiveness of the communication among each subsystem. The backend subsystem was tested starting from the raw recordings collected on-field, then the frame extraction was done and finally the creation of bounding boxes and labels to these frames, as shown in Figure 31. The resulting dataset is composed of more than 5000 frames, highlighting the higher heterogeneity and representativeness. This dataset is used as input for re-training the counting algorithm.

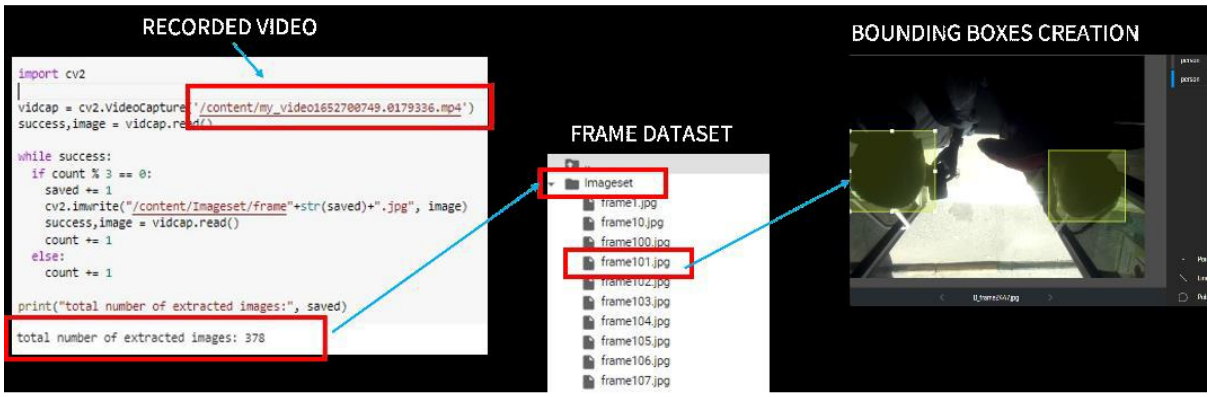


Figure 31. Pre-Processing test

The successful training and testing of the detection, tracking and counting phases is depicted in the sequence of images in Figure 32 and Figure 33.

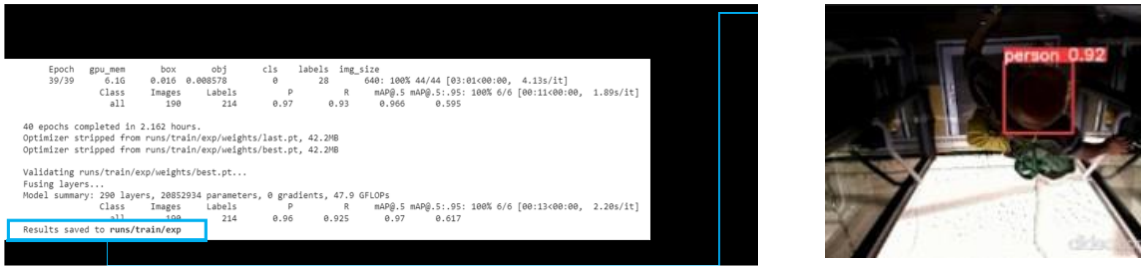


Figure 32 Object detection test



Figure 33 Object tracking and counting test

4.1.3 Performance analysis

This section first presents the results of the accuracy assessment. Secondly, the performances obtained on the experiment for both the commercial system and the low-cost device are compared.

Assessment of Accuracy of Commercial APC

The accuracy of the commercial APC system was found to be 53.17% for boarding and 55.29% for alighting. When resetting the calculated vehicle load once per day, the accuracy of the commercial APC was found to be 21.24%. Yet, the accuracy when resetting the vehicle load to zero at the beginning of each new journey increased to 57.74%.

When looking at accuracy for individual days, as shown in Figure 34, the highest overall accuracy was achieved for 25 October 2021, with a vehicle occupancy accuracy of 60.40%; on that day, the accuracies for boarding and alighting were, respectively, 56.25% and 54.67%. The worst accuracy obtained was for 28 October 2021, with 53.78% for vehicle occupancy, 50.32% for boarding and 51.33% for alighting, as shown in Figure 35.

In some cases, the calculated vehicle occupancy fell below zero due to errors in the numbers of passengers boarding and alighting, as counted by the APC. Overall, the percentage accuracy of the commercial system was between 50% to 65% when considering individual journeys.

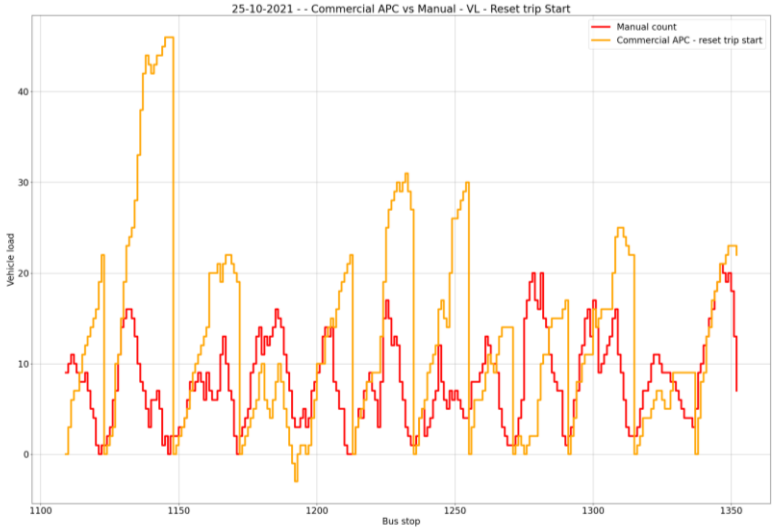


Figure 34. Highest overall accuracy: 25 October 2021

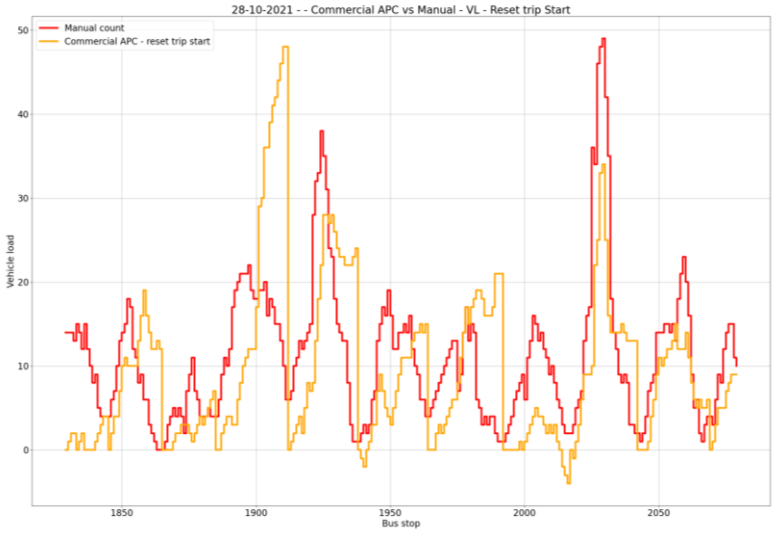


Figure 35. Lowest overall accuracy: 28 October 2021

Results of the Test on Two APC Systems

Although the data collection was carried out over six days, GTT provided data files only for two days, 30 and 31 May 2022, since there had been a malfunction in the commercial APC on the other days. We, therefore, decided to test both systems only for the days for which GTT provided the data. Over the two days, the bus was assigned to two different routes. On day 1 it

was operating on the periphery of the metropolitan area (on what we have called the "uncrowded line"), while on day 2 it was operating on a route (the "crowded line") running through the tourist district and past schools and an outdoor market.

During the analysis, we realized that some data from the commercial APC were lacking for the period in which the manual count was being conducted. Thus, a direct comparison between the commercial and low-cost systems was performed only for day 1, while the low-cost APC was evaluated in relation to the manual count over the whole six days. The accuracy results for the two systems are reported in in Table 19.

Table 19. Accuracy results from both systems

APC System	Boarding	Alighting	Vehicle Occupancy
Low-cost APC system			
Overall accuracy in six-day period	72.27%	74.59%	81.59%
Uncrowded line	83.53%	94.87%	82.46%
Crowded line	65.83%	66.68%	80.38%
Commercial APC system			
Overall accuracy Asti's twenty-day period	53.17%	55.29%	57.74%
Uncrowded line	77.69%	83.33%	50.94%
Crowded line	-	-	

- Low-Cost APC

The overall performance of the low-cost system was much better than expected. The low-cost system achieved an overall accuracy of 81.59% for vehicle occupancy, 72.27% for boarding passengers, and 74.59% for alighting passengers, as reported in Table 19. Furthermore, the distribution of vehicle occupancy was very similar to that found by the manual count (Figure 36).

The performance was better on the uncrowded line, particularly regarding boarding and alighting passengers, with an accuracy, respectively, of 83.53% and 94.87%. On the crowded line during the peak hours, the algorithm had difficulty counting the number of passengers when they were arriving in large numbers. The highest peak observed in Figure 36 corresponds to when the outdoor market was active and when pupils were returning home after school.

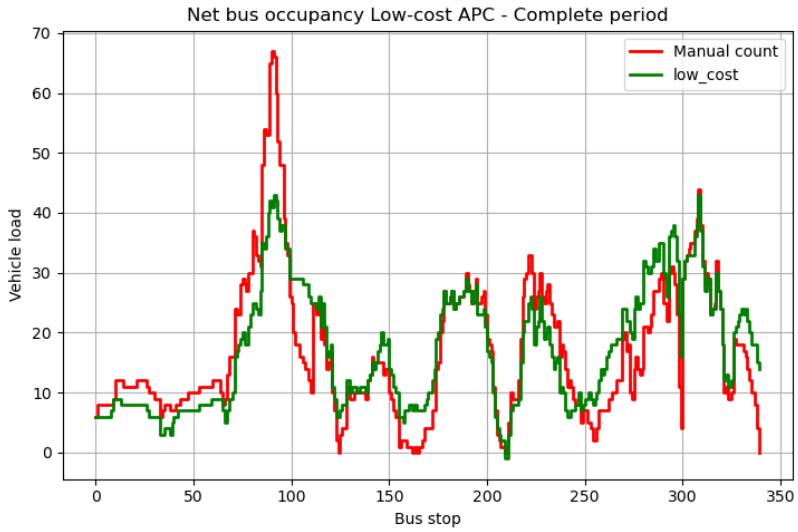
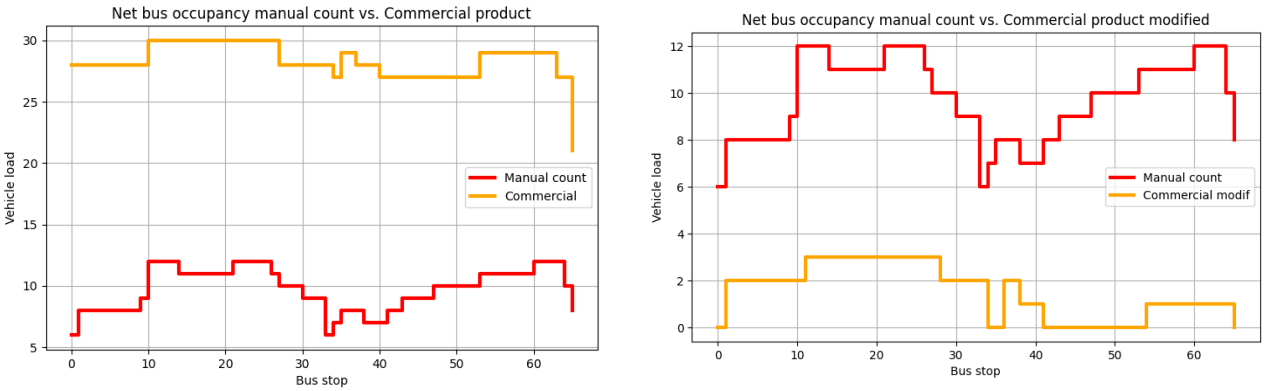


Figure 36. Vehicle occupancy for the low-cost APC system over the whole 6-day period.

- Commercial APC

Due to the lack of data for one of the days, resulting from a malfunction in the commercial APC, we were unable to carry out an evaluation on the crowded line for day 2. On the uncrowded line, the commercial device failed to exhibit the accuracy specified in the product datasheet. As reported in Table 19, the accuracy for boarding and alighting passengers was, respectively, 77.69% and 83.33%.

The accuracy of vehicle occupancy was the lowest, with a value of 50.94%. The poor performance in relation to vehicle occupancy was the consequence of an accumulation of erroneous estimations from previous journeys. For this reason, there was a recalculation of the vehicle occupancy in which the value was reset to 0 when a new journey started. Despite this recalculation, the new vehicle occupancy accuracy was 26.92%. Figure 37 shows the bus occupancy with and without recalculation, and we remark that the original value (before recalculation) is closer to the ground truth pattern distribution.



(a) **(b)**

Figure 37. Vehicle occupancy (a) with original data and (b) with recalculated value

- Comparison between the Systems

The comparative performance of the two systems was evaluated in relation only to the uncrowded line. As shown in Table 19, the overall performance of the low-cost device was better than that of the commercial APC. The low-cost device obtained a considerably higher accuracy for vehicle occupancy (a difference of 31.52 percentage points with the commercial APC), while the percentage point differences for boarding and alighting were 5.84 and 11.54. This trend is also evident in Figure 38, where the vehicle occupancy estimation by the low-cost system closely aligned with the ground truth throughout the evaluation period, while the estimation by the commercial APC was nearly double the value of the manual count, as shown in Figure 39a.

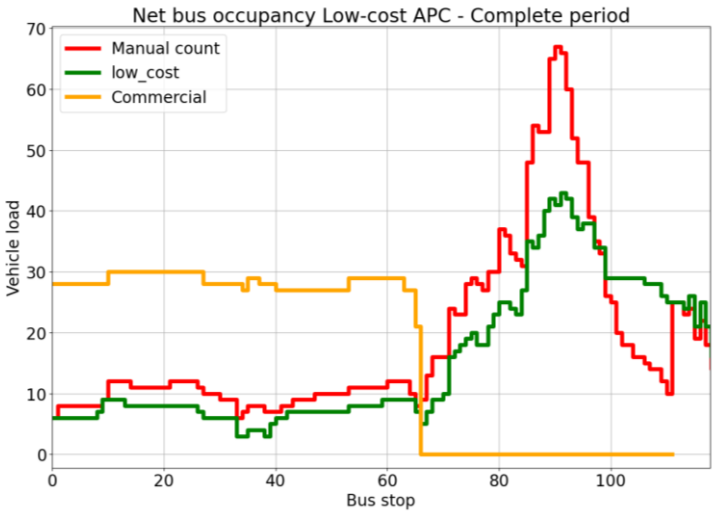
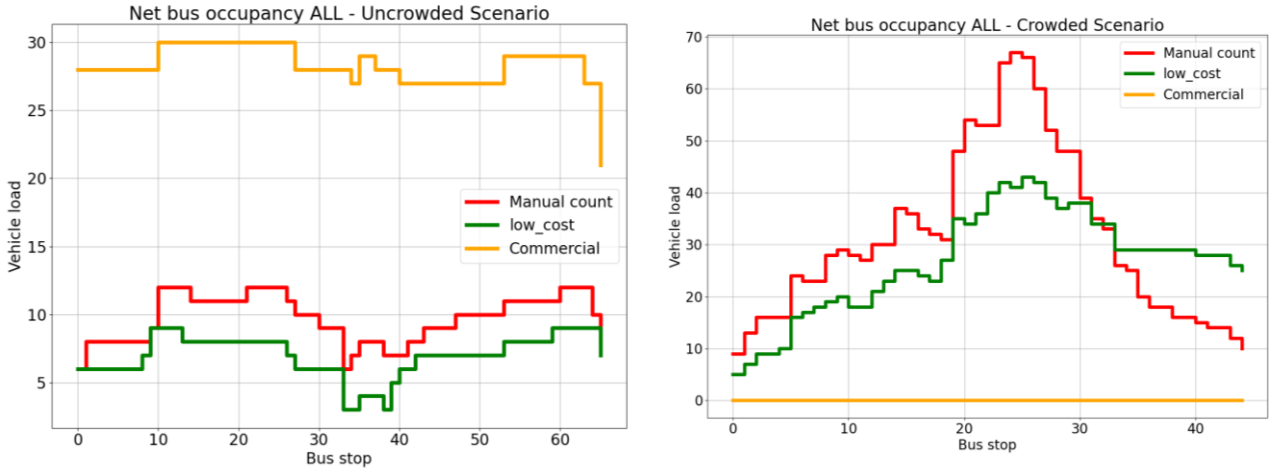


Figure 38. Vehicle occupancy for the three counting systems for the 2-day period



(a) **(b)**

Figure 39. Vehicle occupancy for the three counting systems on (a) the uncrowded line and (b) the crowded line.

From Figure 39 we see that the low-cost system has a closer correspondence to the ground truth in both scenarios, confirmed by the correlation values. The highest correlation (0.88) is for the vehicle occupancy obtained by the low-cost system. The lowest correlation (0.062) is for the alighting passengers counted by the commercial APC. Where there is a strong correlation with the manual count, the collected data provide an opportunity for improving the accuracy of the estimations via a correction model. The values estimated by the low-cost system are such that a correction model may potentially be used to advantage in making them more reliable.

Conclusion pilot study

The study was conducted in two steps: (1) ground truth and APC data were collected over a period of 20 days in Asti to assess accuracy, and (2) a low-cost APC system was developed and implemented, followed by 6 days of data collection on a bus operating in Turin. The performance evaluation made in Asti shows that the estimation of the accuracy of vehicle occupancy of the commercial APC system, after some post-processing, reached a value of 57.74%. For boarding and alighting, the accuracy was found to be below 55%. The low-cost APC system, however, during the 6 days of data collection in Turin, achieved markedly better results. The accuracy of vehicle occupancy was 81.59%, while the average accuracy for boarding and alighting was 74%. The results show that the overall accuracy of the low-cost system was higher by approximately 20 percentage points compared to the commercial APC system assessed in Asti.

The data collected over 6 days in Turin were limited due to malfunctions in the commercial APC system, resulting in only one day of data from the uncrowded line being provided by the transport company. With the available data, the results show that the low-cost device had better

accuracy overall than the commercial APC already installed on the bus. The vehicle occupancy estimated by the low-cost system had an accuracy of 82.46%, while that estimated by the commercial APC had an accuracy of 50.9%. For boarding and alighting, the low-cost device had accuracies, respectively, of 83.5% and 94.8%, while the commercial APC had accuracies of 77.6% and 83.33%. Although a direct comparison between the two APC systems in Turin under crowded conditions was not feasible, valuable insights can be extracted from the data gathered in Asti concerning the performance of the commercial APC system on crowded lines. As the number of passengers of public transport in Turin is higher, being a bigger city than Asti, we can presume that the accuracy in Turin could potentially be lower.

Regarding the performance, the low-cost system had the lowest accuracy during peak hours, more obviously in the case of a crowded line, while the commercial system had a very low accuracy in relation to occupancy since it accumulated boarding and alighting errors along the route. Our findings complement those from the literature, confirming that there is a higher error level for net bus occupancy during peak hours (Grönman *et al.*, 2019; Hsu *et al.*, 2020a; Olivo *et al.*, 2020; Mccarthy *et al.*, 2021).

The results indicate that there is a potential for using low-cost APC systems, which represent a good compromise between cost and performance However, there are technicalities to be addressed before this system may be officially adopted. Certified compliance with a number of licenses and standards is required for any device implemented in a public transport vehicle, and consideration is currently being given to a possible future integration of the low-cost system with the transport company's existing GTFS. Ultimately, all this would have a cost that has not, so far, been included in the cost of our device.

Additionally, the implementation of a secure protocol for transmitting videos or frames over the network is necessary; the computational power of the RaspberryPi 3B is not enough to run an object detection and tracking algorithm smoothly without encountering functional issues. Also, the collected dataset could be utilized to improve the estimation of the accuracy of APC during a post-processing phase (Mccarthy *et al.*, 2021). Nevertheless, our experiment highlights the potential of deep learning models in improving the accuracy of both real-time and post-processing schemes. This is, to our knowledge, the first study to demonstrate the potential of a low-cost APC system by comparing it with a traditional APC system in a real-world setting.

This experiment also raises concerns about the performance of current commercial APC systems. Although the formula used to evaluate accuracy was not the formula specified in VDV recommendation 457-2, there was a discrepancy between the measured accuracy (57.74%) of the commercial APC and the accuracy claimed on the product's datasheet (see *Table 1*). By comparing the two APC systems, it is evident that the estimation of the commercial APC system is susceptible to many variabilities; from the development of the low-cost system and the subsequent collection of ground-truth data, we could understand and postulate that the video-based APC systems can be affected by the following external factors: (i) the boarding or alighting of large group of passengers may cause over- or under-counting (Liciotti *et al.*, 2017; Mccarthy *et al.*, 2021; Amato *et al.*, 2022; Seidel *et al.*, 2022); (ii) passengers who step off

(alighted) the bus and immediately re-board could lead to duplicated counts (Grönman *et al.*, 2019); (iii) when the bus reaches the full capacity, the system faces difficulty in accurately counting passengers due to the crowd around the door (Hsu *et al.*, 2020b); (iv) passengers lingering at the door during the entire trip can lead to inaccuracies (Seidel *et al.*, 2022); (v) technical settings such as positioning (Grönman *et al.*, 2019; Mccarthy *et al.*, 2021), resolution quality (Liu *et al.*, 2017), lens cleanliness and device calibration of the device can impact accuracy; and (vi) the possibility for the commercial APC system to miss some passengers because it is switched off or re-started at each terminus when the bus stays idle. Factors like lighting conditions have affected the results in some studies (Liu *et al.*, 2017; Grönman *et al.*, 2019; Hsu *et al.*, 2020a; Seidel *et al.*, 2022), though the results of our model were not significantly affected by different lighting conditions. Similarly, the presence of large elements (Seidel *et al.*, 2022) such as bicycles or baby carriage, or wearable objects (Kniess *et al.*, 2021), i.e., hats or backpacks, did not significantly affect the estimation of our low-cost APC system. This could be attributed to the use of multiple frames in our training data that included different lighting conditions and types of elements.

Despite our limited knowledge of the manufacturers' image processing algorithms (not disclosed), our hypothesis suggests that sub-optimal performance may be related to insufficient updates to their estimation software and a lack of proper maintenance. An additional plausible factor, as observed during our ground-truth data collection, could be attributed to the positioning of the commercial APC system in proximity to the bus door. This particular area of the bus exhibited greater vibrations on poor road surfaces compared to the ceiling-mounted location of the low-cost system. Hence, this study shows that it would probably be worthwhile to invest in improving the accuracy of the commercial APC systems, be it through improvements to image processing algorithms or evaluation about placement of the cameras inside the bus. The parameters of occupancy, boarding and alighting are very important because they are key indicators of the performance of the transport networks. By ensuring that these values are in line with actual scenarios, the quality of service can be significantly improved. Therefore, we believe that transport companies and APC system manufacturers should invest in improving the performance of existing systems.

Future work should, therefore, be devoted to exploring the significant potential for increasing the estimation of APC accuracy. Given the impressive rate at which object recognition and tracking algorithms are being improved, there is considerable potential for increasing estimation accuracy by applying and testing algorithms developed in the literature in actual in-field experiments. Investigations into the placement of the cameras within the bus and their susceptibility to vibrations might also be conducted, exploring their impact on the accuracy of passenger counting. This analysis could aid in determining the most suitable camera placement.

In addition, new hybrid methodologies might be explored with a view to developing more robust low-cost real-time passenger counting systems. One such approach could involve the implementation of edge-cloud object recognition models, where a portion of the estimation occurs onboard while another part takes place in the cloud (Oroceo *et al.*, 2022). An alternative

strategy involves the testing and adoption of various security protocols for transmitting frames or videos. This aims to leverage superior external computational power (cloud) and consequently attain more precise estimations (Fort *et al.*, 2021). Future work could also focus on evaluating other types of commercial APC systems and testing them against a low-cost system in real-world conditions under different crowding scenarios for longer periods.

4.2 Data fusion

Data fusion involves integrating multiple data sources to improve the accuracy and reliability of the resulting dataset as explained in the methodology. This section delves into the results of various techniques used for feature selection and model tuning, which are critical steps in the data fusion process. Then, the results for the different models in different contexts are presented and compared.

4.2.1 Feature Selection

There were multiple different combinations of feature selection and ranking attempts made. Some combination include – a) Encoded and Non-normalised DB, b) Encoded and Normalised DB using just select K-BEST for mutual info regression. Figure 40 and Figure 41 show the importance of different features.

Then a combined approach was used – where four different algorithms were used to calculated feature importance score and the sum of the scores as well. The results are shown first in the plot reported in Figure 42. As we can see from the figures, apart from the VL_GTT variable, the target encoded stop IDs are getting high scores. This could mean that there is a pattern of variability based on current and next stop Ids on the crowding levels and, with sufficient data, this could be a valuable input for a data fusion algorithm. The number of records collected in this study may not be enough to learn these patterns for all stops in the city of Turin but considering that the manual counting was focused on a few specific lines, a decision was made to keep these features.

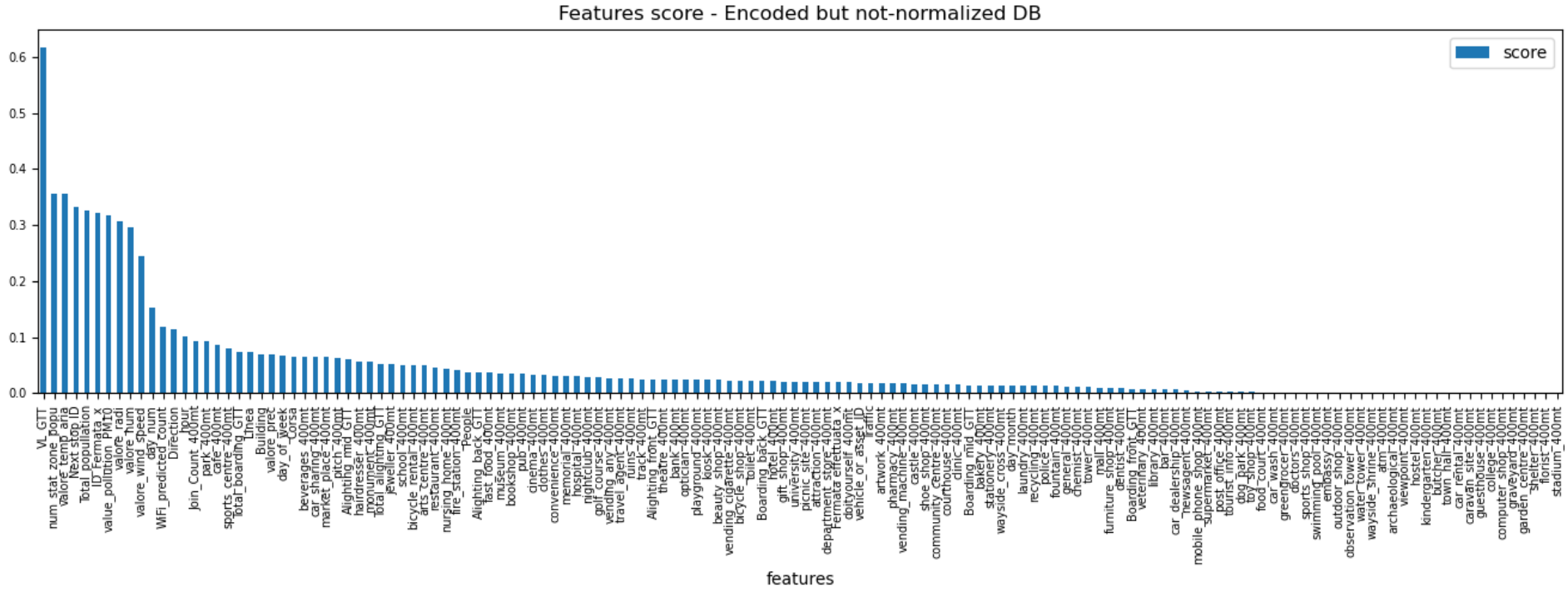


Figure 40 Feature selection - encoded but non-normalised DB - mutual info scores

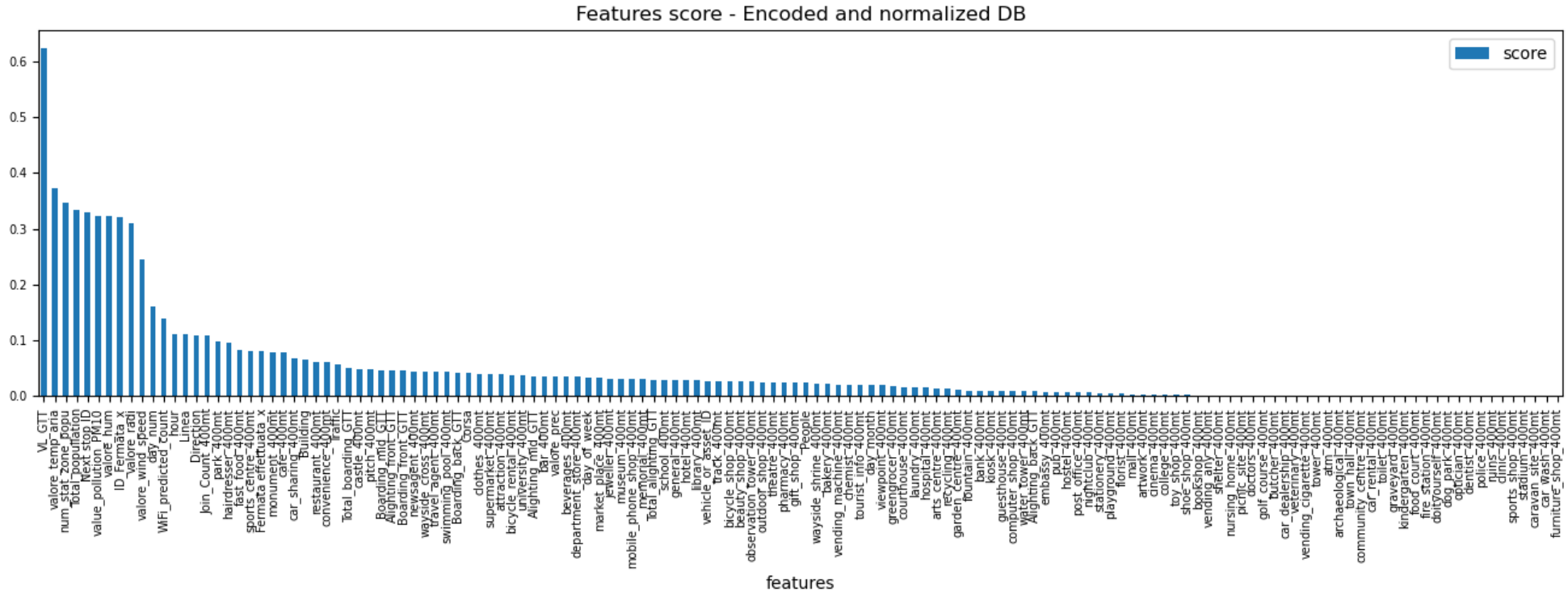


Figure 41 Feature selection - encoded and normalised DB - mutual info scores

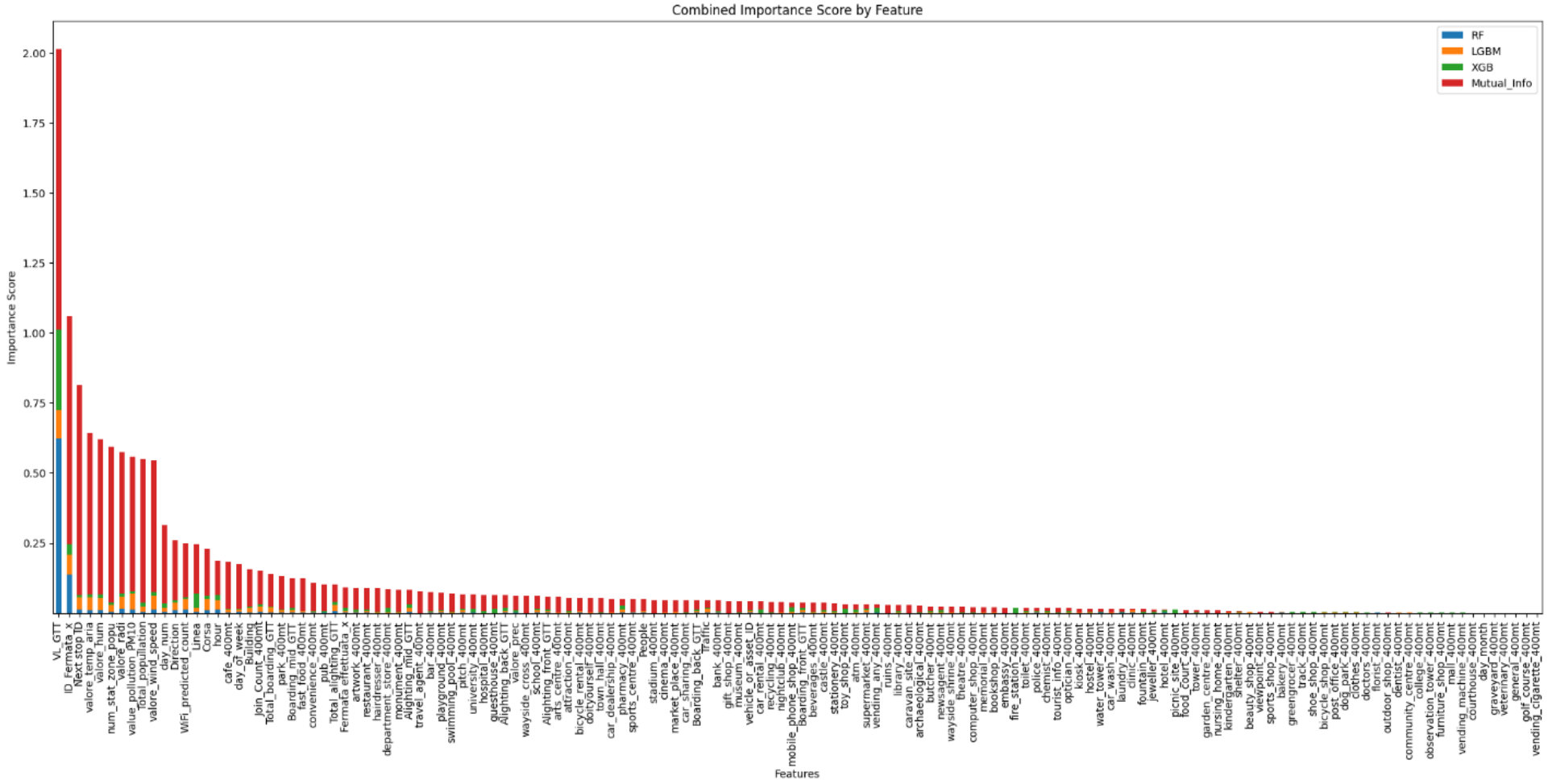


Figure 42 Feature Importance - Combined score - 4 methods

The Table 20 contains the data for the feature importance and scores them using four different methods and sorted by the sum score which is a sum of the importance score from the four different methods. The results shown here refer to the encoded and normalised DB. Only the top 20 features based are shown here.

Table 20 Feature Importance scores - different methods and sum score

features	RF	LGBM	XGB	Mutual_Info	Sum_Score
VL_GTT	0.627052099	0.107333333	0.267051846	1	2.001437279
ID_Fermata_x	0.13411613	0.077	0.034298472	0.825483995	1.070898597
Next stop ID	0.012773111	0.044	0.008723033	0.739164858	0.804661001
valore_temp_aria	0.010818062	0.042666667	0.01036658	0.57929734	0.643148648
valore_hum	0.011378997	0.041333333	0.01234888	0.517995608	0.583056818
valore_radi	0.0155433	0.043666667	0.009669479	0.494072851	0.562952297
num_stat_zone_popu	0.006503599	0.019333333	0.009199779	0.521670535	0.556707245
value_pollution_PM10	0.011614278	0.054666667	0.007300039	0.479618995	0.553199979
valore_wind_speed	0.013209681	0.052	0.011822755	0.4300443	0.507076736
Total_popullation	0.005156478	0.021	0.014075207	0.46268483	0.502916515
day_num	0.005270433	0.013	0.015865136	0.259707147	0.293842716
hour	0.013095725	0.036666667	0.018247167	0.200242192	0.268251751
Direction	0.011556997	0.023333333	0.01037572	0.212560754	0.257826805
WiFi_predicted_count	0.010964247	0.041666667	0.005624115	0.170316082	0.228571111
Linea	0.006214997	0.008666667	0.077430911	0.134778267	0.227090842
Corsa	0.010292172	0.038	0.009622645	0.168028836	0.225943653
Join_Count_400mt	0.004237927	0.021	0.008840436	0.136086727	0.17016509
park_400mt	0.002776277	0.007666667	0.002980426	0.111747629	0.125170999
cafe_400mt	0.001567527	0.007	0.002042824	0.105897344	0.116507695
fast_food_400mt	0.001645551	0.006	0.001977985	0.101530133	0.111153669

After the above analysis, a check was made to select only features that are in the top 20 features for the four different methods and also the top 20 for the two list of sum scores (non-normalised and normalised). This choice of 20 was also based on taking a look at the feature importance scores on a plot; 0.01 using RF scores and 0.15 for mutual info were identified as potential elbow points. There were 17 features that satisfied this condition of being in the top 20 in all the lists and they are listed in Table 21. Any feature that did not have 4 in the first column and 2 in the second column was not considered.

Table 21 Instances of each feature in the Top 20 ranking for different feature selection algorithms

Feature	Top 20 instances - all 4 methods	Top 20 instances – using only Multi-score
VL_GTT	4	2
ID_Fermata_x	4	2
Next stop ID	4	2

valore_temp_aria	4	2
valore_hum	4	2
valore_radi	4	2
num_stat_zone_popu	4	2
value_pollution_PM10	4	2
valore_wind_speed	4	2
Total_popullation	4	2
day_num	4	2
Hour	4	2
Direction	4	2
WiFi_predicted_count	4	2
Linea	4	2
Join_Count_400mt	4	2
cafe_400mt	4	2

4.2.2 Model tuning

The performance of various models was evaluated based on accuracy, mean absolute error, and correlation coefficients. The tuning results for each model are summarised below.

The Prophet model was tuned, and the best results were obtained with daily seasonality set to True and a multiplicative seasonality mode. Additionally, a period set to 1/24 (hourly seasonality) and a Fourier order of 3 provided the best performance. The performance for some configurations with the best results are presented in Table 22.

Table 22 Hyperparmater tuning results for Prophet model - select few rows

Period	Daily seasonality	Seasonality mode	Fourier order	Percentage accuracy	Spearman corr
1/24	*TRUE*	*multiplicative*	*3*	*70.68025954*	*0.610302675*
1/24	TRUE	*multiplicative*	9	70.63380974	0.597599533
1/96	TRUE	*multiplicative*	15	70.57060926	0.606748146
1/96	TRUE	*Additive*	21	70.5317266	0.601782574

The best SARIMAX parameters identified were order: (1, 0, 0) and seasonal_order: (0, 0, 0, 24). The SARIMAX model with order: (1, 0, 0) and seasonal_order: (0, 0, 0, 24) achieved a percentage accuracy of 72.13%, a mean absolute error of 6.16, a Pearson correlation of 0.363, and a Spearman correlation of 0.416. This model was also identified as the best fit by the auto_arima function based on the Akaike Information Criterion (AIC).

A hyperparameter tuning process was conducted using K-Fold cross-validation with five splits to identify the best combination of parameters. The cross-validation process aimed to minimise the validation mean squared error (MSE). The optimal hyperparameters identified for Bayesian regression were determined to be:

- alpha_1 = 1e-06
- alpha_2 = 1
- lambda_1 = 1
- lambda_2 = 0.001.

The XGBoost and LightGBM models were tuned using RandomizedSearchCV function with a mean squared error loss and the best results were obtained with the following parameter values:

- **XGBoost Parameters:**
    - n_estimators=1000
    - n_jobs=-1
    - max_depth=20
    - learning_rate=0.1
- **LightGBM Parameters:**
    - n_estimators=1000
    - n_jobs=-1
    - max_depth=20
    - learning_rate=0.1
    - num_leaves=50.

LSTM

For LSTM, hyperparameter tuning was done utilising Hyperband tuner from keras_tuner library. The tuner identified the model parameters that minimised the loss function; lower the score, the better is the model. The results from some of the hyperparameter combinations with the best results are listed in Table 23

Table 23 Hyperparameter tuning results for LSTM - select few rows

units	dropout_rate	optimizer	loss	score
150	0.2	adam	mean_squared_error	0.094283216
150	0.3	adam	mean_squared_error	0.094479084
100	0.4	adam	mean_squared_error	0.094517551
100	0.3	adam	mean_squared_error	0.09459392

4.2.3 Performance evaluation

After the model tuning, the performances of the different models were evaluated with the best parameters found during the hyperparameter tuning phase. As mentioned in the methodology, the models are evaluated on three different datasets after using imputation. The breakdown for each dataset is as follows.

The SIMPLE dataset, resampled to every 30 seconds, strictly includes data only when available (e.g., data are available at two consecutive stops, but not for a timestamp in between, assuming vehicle load does not change between stops). The record counts are as follows:

- Training records: 4036
- Validation records: 448
- Testing records: 1923
- Total records: 6407.

The KNN_FULL dataset has the imputation algorithm applied to the entire dataset. The record counts are as follows:

- Training records: 8952
- Validation records: 995
- Testing records: 4263
- Total records: 14210.

The KNN_TRAIN_ONLY dataset has training and validation records as in KNN_FULL and test records as in SIMPLE. The record counts for KNN_TRAIN_ONLY are as follows:

- Training records: 8952
- Validation records: 995
- Testing records: 1923
- Total records: 11870.

As can be seen from the dataset sizes, there were a large number of missing records when the dataset was created based on all timestamps within the earliest and latest times for which data were available on any day (08:53 AM to 05:20 PM), to enable the models to learn some seasonality. It should be noted that data were collected by the author for 57 days, but one of the main inputs for the data fusion approach – the camera-based APC system data from GTT (Turin's public transport operator) – was provided only for 14 days. This severely compromised the amount of data available and resulted in not only a very high percentage of imputed data but also potentially very few samples for some time ranges, which could lead to presenting a picture that is not very close to reality. However, the results are compared for all three datasets, using only those 14 days and the GTT dataset as the benchmark for accuracy, to see if data fusion techniques improve the overall performance.

Simple Database

In the following section, the results are presented in a structured manner for each of the three datasets. Initially, a table containing the accuracy metrics for all the attempted models, along with the accuracy metrics of the benchmark (GTT), is provided. Following this, scatter plots illustrate the performance of the different algorithms against the ground truth values. Subsequently, sample line graphs demonstrate the performance of different algorithms against the ground truth data. These line plots are categorised into three sets: one set depicting a high level of accuracy, another set showing some of the worst performances of the models, and a

final set highlighting some anomalies. The accuracy metrics results for the simple database are first presented in Table 24.

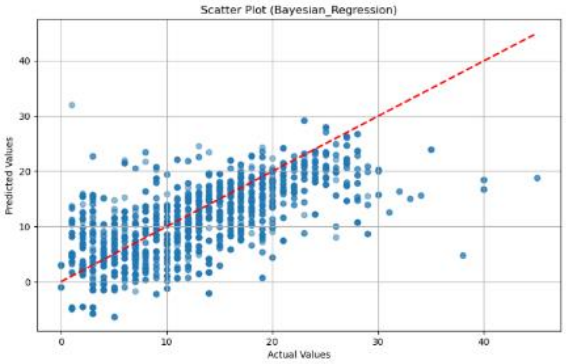
Table 24. Performance evaluation - Simple dataset

SIMPLE DATABASE				
model	**Accuracy (%)**	**MAE**	**pearson_corr**	**spearman_corr**
XGBoost Test	77.66494809	4.55	0.70089468	0.721301517
LightGBM Test	76.25944678	4.43	0.759806096	0.777143729
LSTM	75.96313559	4.8	0.648825295	0.645445205
SARIMAX	75.50881582	5.24	0.573341775	0.605084709
Bayesian_Regression	75.02374501	4.49	0.670021237	0.688265292
Prophet	70.49138654	5.29	0.583634169	0.61082323
GTT	62.05366626	5.36	0.659796819	0.69221466

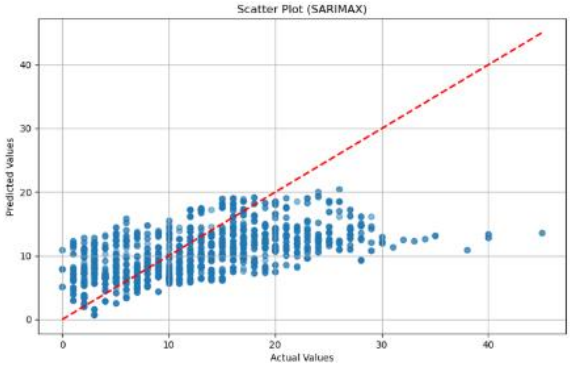
For the simple dataset, the best accuracies and correlations were achieved using the two machine learning algorithms, XGBoost and LightGBM. LightGBM recorded a lower error and better correlation, while XGBoost showed a higher percentage accuracy. Overall, all the algorithms in this dataset performed better than the benchmark – both in terms of percentage accuracy and MAE – which is the accuracy achieved by the GTT data. The scatter plots shown in Figure 43 for the simple dataset reveal that the machine learning algorithms, particularly XGBoost and LightGBM, exhibit the best performance, with tight clustering around the red line indicating high prediction accuracy and low error rates. Bayesian Regression and LSTM show moderate performance, with reasonable but less accurate predictions. In contrast, SARIMAX and Prophet FB display significant spread and deviation, indicating lower prediction accuracy and reliability. The GTT benchmark underperforms compared to all the tested algorithms, highlighting the effectiveness of the data fusion models in improving prediction accuracy and reducing error.

Examining the line plots, Figure 44 displays a sample section of the test dataset where most algorithms performed well. The plots show that nearly all algorithms closely track the truth values, with XGBoost performing the best. It is evident that Prophet and Bayesian Regression share a false peak around samples 3-4, which was also present in the GTT dataset; however, other algorithms have corrected this anomaly, presumably using data from other features. Additionally, there is a slight misalignment in the LSTM plot due to the nature of how input data for LSTM is processed in sequences with a sliding window approach. This approach results in data points being misaligned by one sequence length (60 in this case). Another characteristic of this sliding window approach for LSTM is the smooth nature of its line plot for predictions, which contrasts with the plots of the other models.

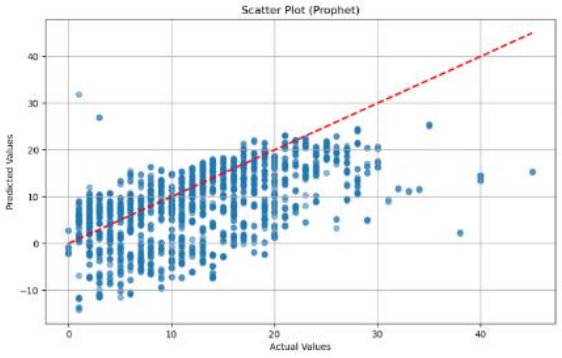
Looking at the plots in Figure 45, it can be observed that in the second half of the plot, the estimated values for almost all models are higher than the actual values. This inflated estimation is not present in the GTT dataset, indicating that it is likely influenced by data from the other features. Conversely, in the first 20% of the same plot, the models have performed better than the GTT benchmark. Figure 46 shows some anomalies in the data. For the GTT source data, a sudden drop in estimated passenger count can be noticed close to sample 50, and this has also affected the models. Similarly, another anomaly is the fact that except XGBoost and LightGBM, all the other models have not tracked the peak in the middle of the plot very well. The disparity in performance across the models can be attributed to their inherent capabilities and limitations in handling non-linear relationships, sudden changes, and feature interactions within the dataset. LightGBM and XGBoost's ability to handle complex patterns and feature interactions likely led to their superior performance in capturing the peaks. LSTM should theoretically well handle sequences and time dependencies due to their recurrent nature. However, in this case, LSTM has not tracked the peaks effectively, which could be due to insufficient data.



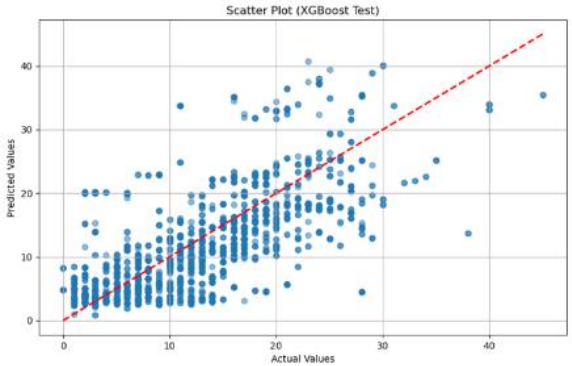
(a) Bayesian Regression



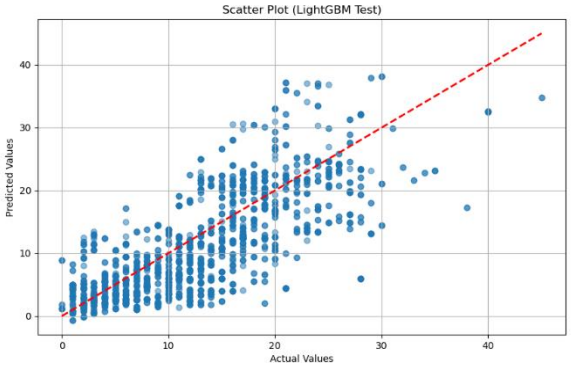
(b) SARIMAX



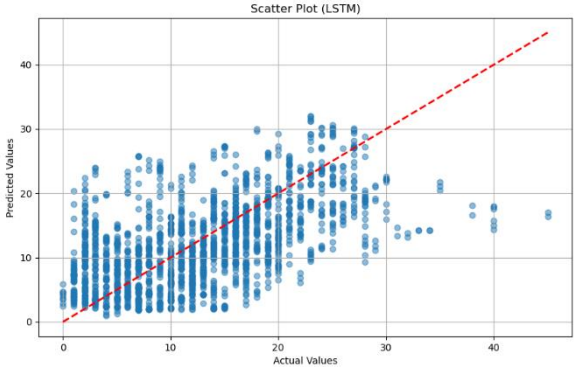
(c) Prophet FB



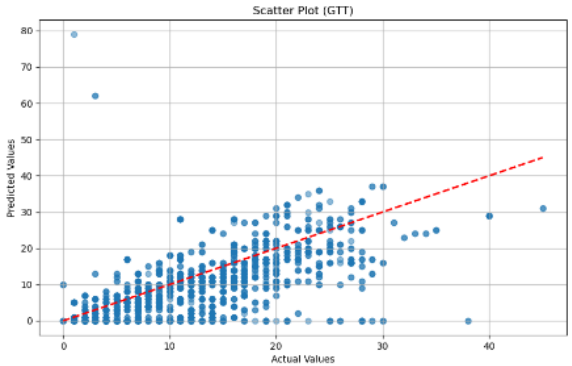
(d) XGBoost



(e) LightGBM

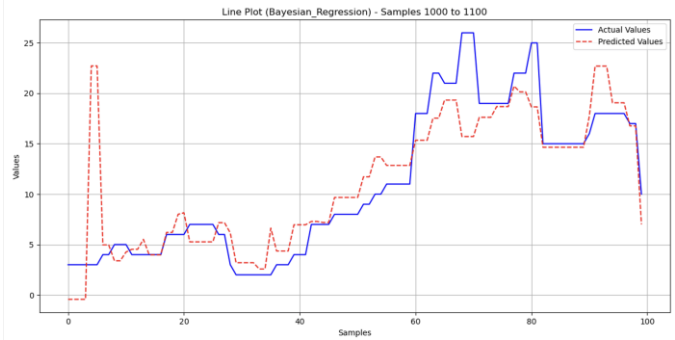


(f) LSTM

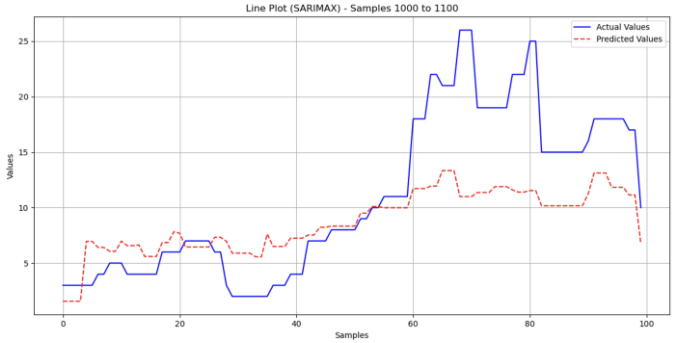


(g) GTT

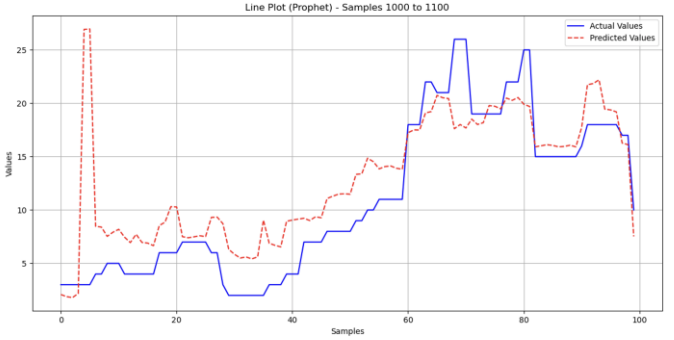
Figure 43. Scatterplot Simple Dataset



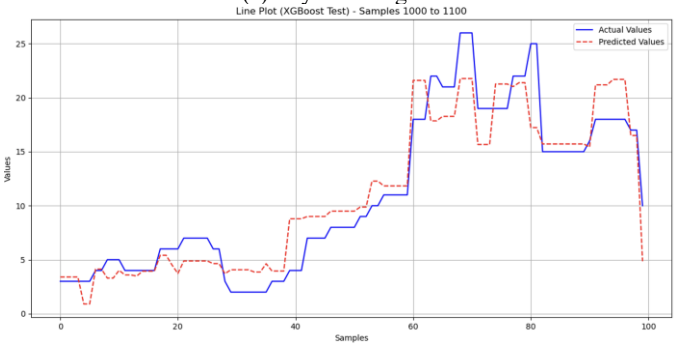
(a) Bayesian Regression



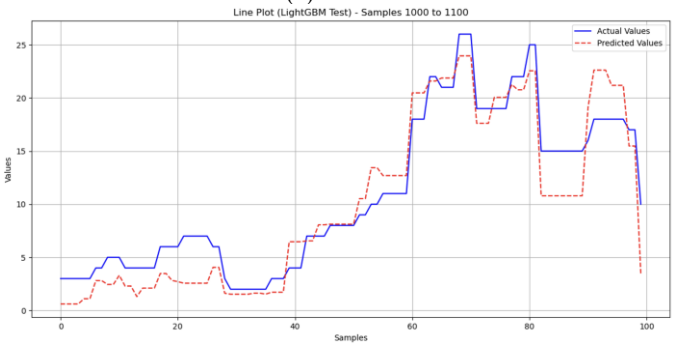
(b) SARIMAX



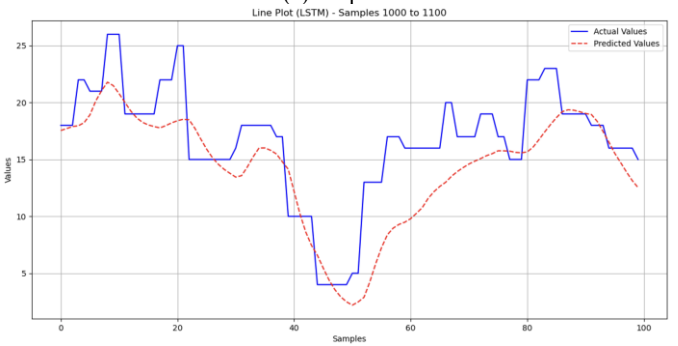
(c) Prophet FB



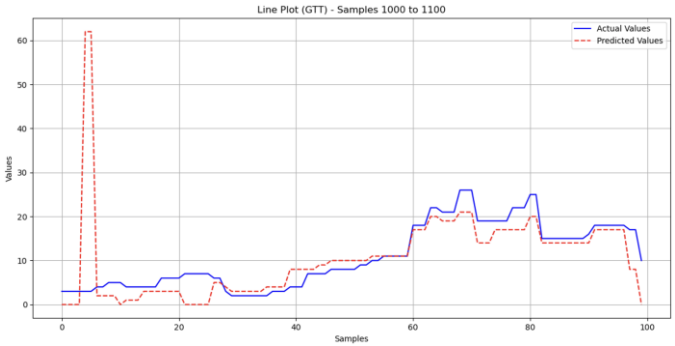
(d) XGBoost



(e) LightGBM

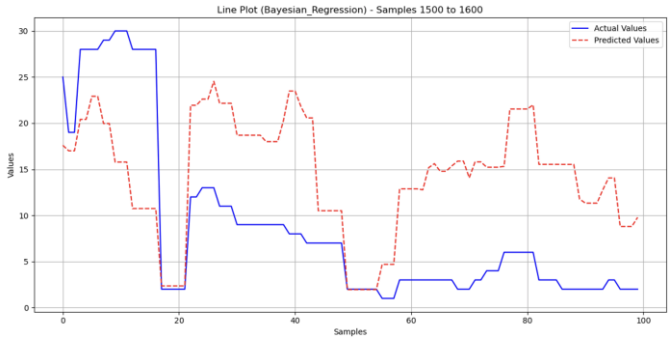


(f) LSTM

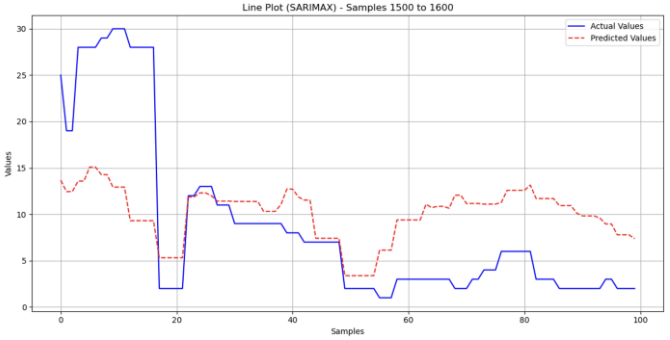


(g) GTT

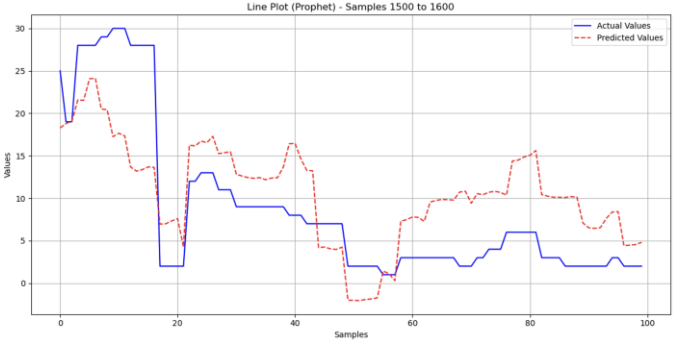
Figure 44. Best datafit graph - XGBoost



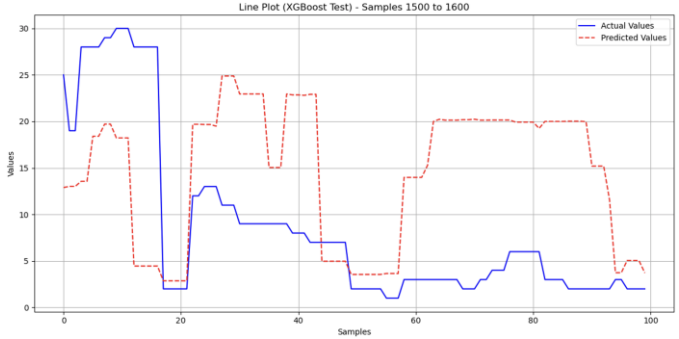
(a) Bayesian Regression



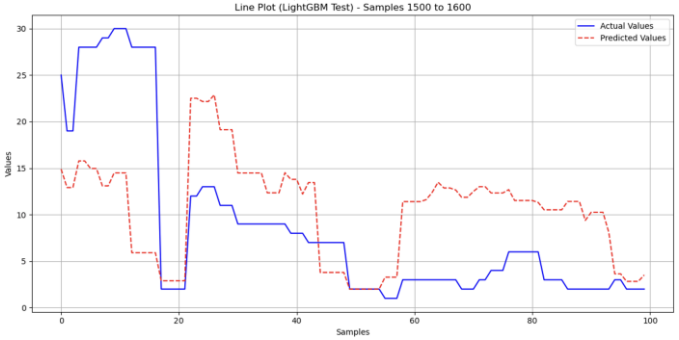
(b) SARIMAX



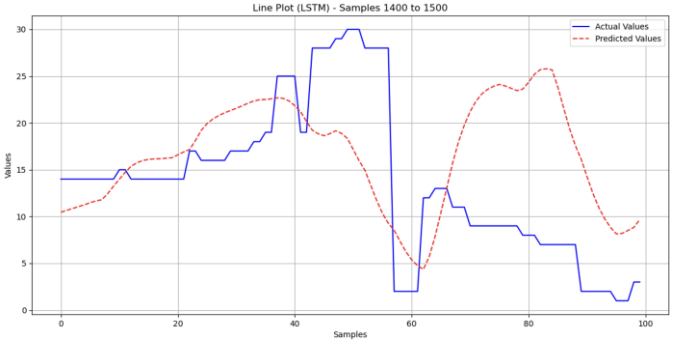
(c) Prophet FB



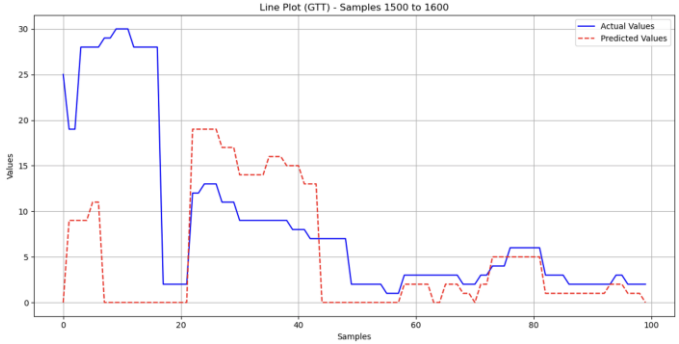
(d) XGBoost



(e) LightGBM

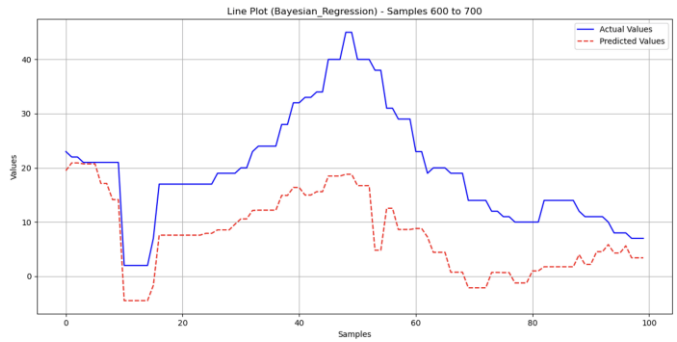


(f) LSTM

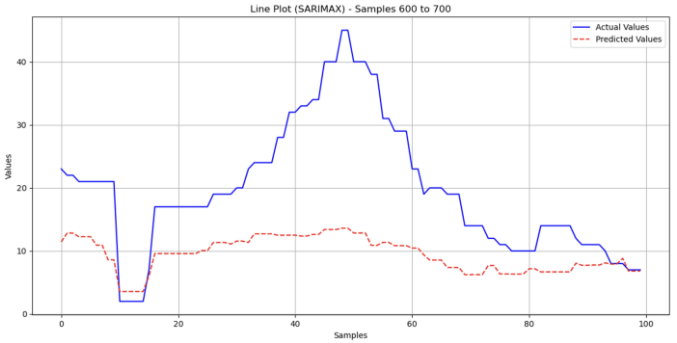


(g) GTT

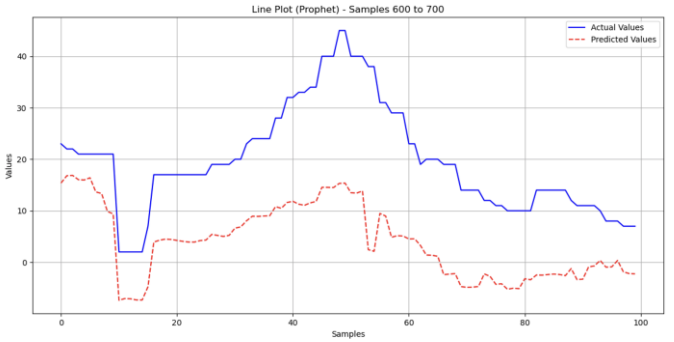
Figure 45. Worst graphs fit - XGBoost



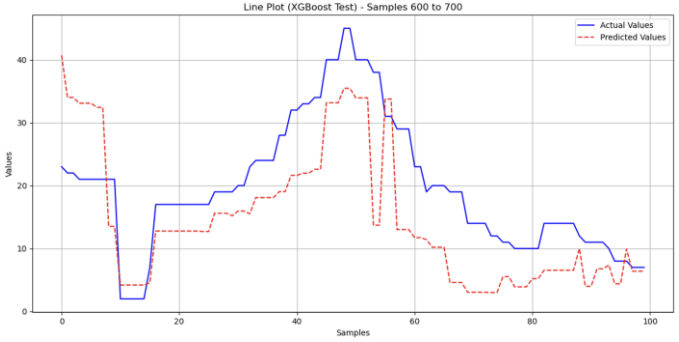
(a) Bayesian Regression



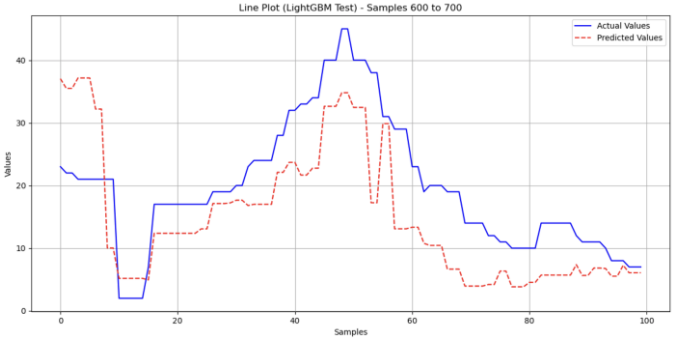
(b) SARIMAX



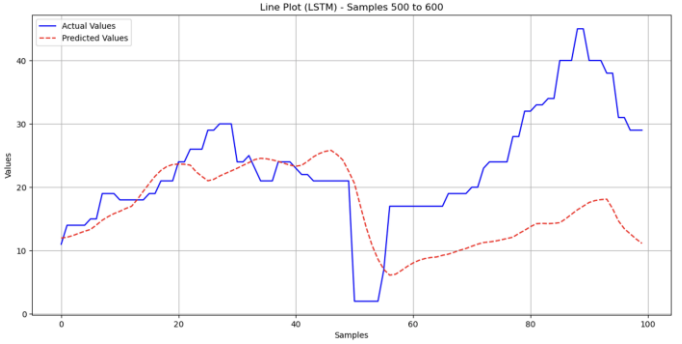
(c) Prophet FB



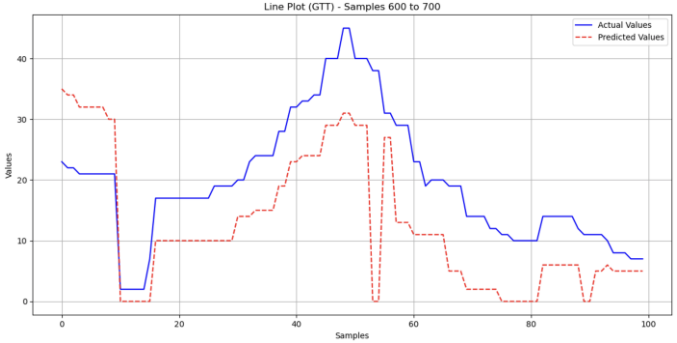
(d) XGBoost



(e) LightGBM



(f) LSTM



(g) GTT

Figure 46. Anomaly behaviour graphs – Simple dataset

Full imputation database

Table 25 demonstrates the accuracy metrics obtained by applying the data fusion methods on the full imputation database. As noted earlier, due to the amount of missing data, the augmented or synthetic data for this dataset form a significant portion of the dataset (more than 50%). Therefore, the results obtained with this dataset should be interpreted with caution. At first glance, this table shows great performance for almost all the models but, considering that the performance of the benchmark GTT dataset has also increased significantly, most of the improvement in performance could be attributed to the augmented data having higher accuracy. An interesting point to note is that only for SARIMAX and Prophet the Pearson correlation has improved considerably, from 0.573 and 0.583 to 0.700 and 0.701, respectively. This could mean that the filling of data has helped the two algorithms that rely the most on seasonality, benefitting from having continuous time series data with some seasonality, even if a significant portion of it was augmented data relying on imputation techniques.

Table 25. Performance evaluation - Full imputation dataset

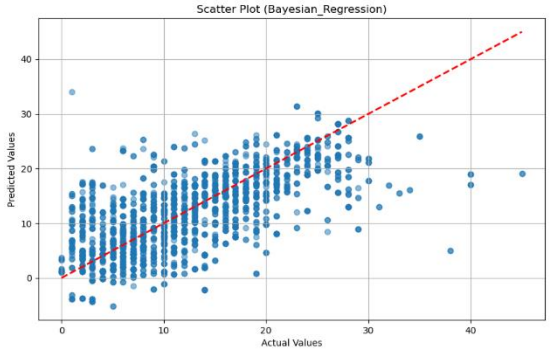
FULL IMPUTATION DATASET				
model	**Accuracy (%)**	**MAE**	**pearson_corr**	**spearman_corr**
LightGBM Test	89.61480801	2.01	0.781995345	0.757142413
XGBoost Test	89.45676401	2.18	0.731231247	0.728569365
Bayesian_Regression	87.32632335	2.38	0.686972428	0.634365758
LSTM	87.20196708	2.82	0.688119097	0.647033686
SARIMAX	84.35799383	3.88	0.700158541	0.634593783
Prophet	77.85910814	6.85	0.701058693	0.644945844
GTT	76.44410736	4.04	0.695599085	0.728278798

The scatter plots for all the models are shown in Figure 47. Scatterplot Full imputation Dataset are very similar to the scatter plots for the simple dataset. Notable differences are observed for the Prophet and SARIMAX algorithms, which exhibit better alignment with the reference line. Additionally, as the dataset size increases, the dispersal of points in the LSTM model becomes more compact and closer to the reference line. The augmentation of the dataset has clearly helped these three algorithms.

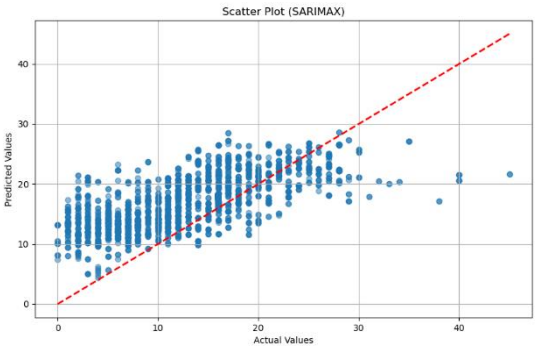
Among the line plots, for the best plots shown in Figure 48, it is evident that the LightGBM algorithm tracks the ground truth line more closely than the other algorithms. In this sample set, LSTM has performed significantly worse than the others. LightGBM, XGBoost, Bayesian, and Prophet algorithms have also effectively handled some anomalous drops in the GTT set. On the other hand, in Figure 49, which represents a portion of the dataset where most algorithms performed poorly, notable trends can be observed. Algorithms like XGBoost,

LightGBM, and SARIMAX underestimated the counts in the first part and overestimated them in the second part. In contrast, Bayesian and LSTM performed well in just the second part, while Prophet showed good performance in the first part of the graph. In general, the “best” plots look better than the same plots in the simple dataset and the “worst” plots look worse than the simple dataset as well.

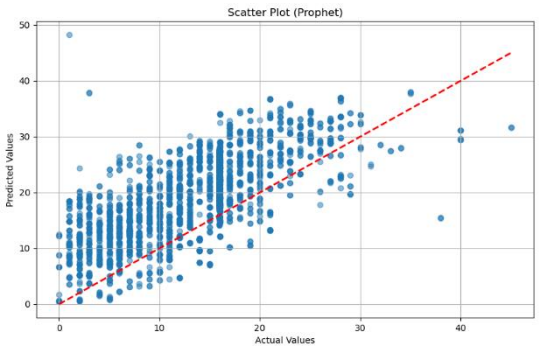
Figure 50 shows some anomalies that were noticed in the plots. Unlike the anomalies noticed in the simple dataset, the anomalies here cannot be attributed to the GTT dataset. Almost all the models at the beginning of the dataset have overestimated the ground truth but the GTT dataset has actually underestimated the counts. On reviewing the data, the other features had high values, specially building density and also the current and next stops had relatively high values. This could potentially be the reason for the overestimation of the counts in this specific sample dataset. While these factors do affect passenger occupancy, they may not always have the same impact and this anomalous behaviour could be attributed to not having enough data for the algorithms to generalise well.



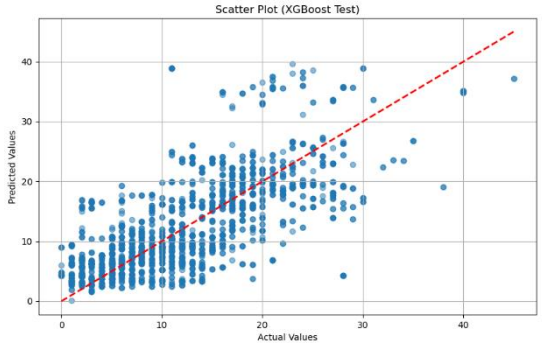
(a) Bayesian Regression



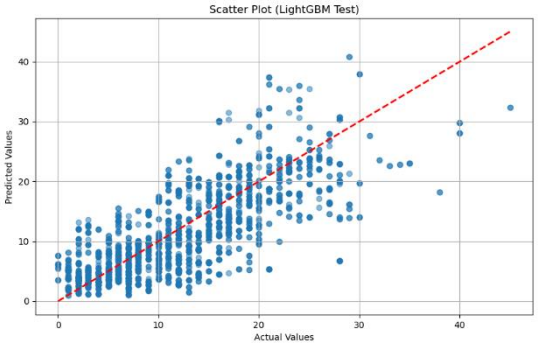
(b) SARIMAX



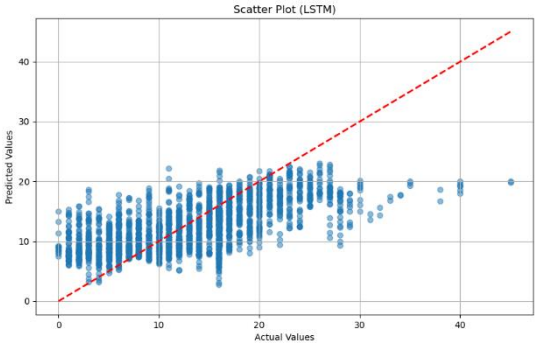
(c) Prophet FB



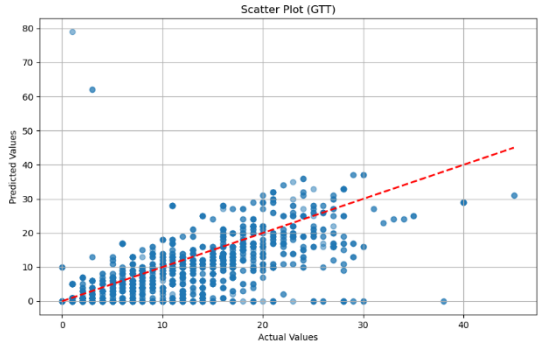
(d) XGBoost



(e) LightGBM

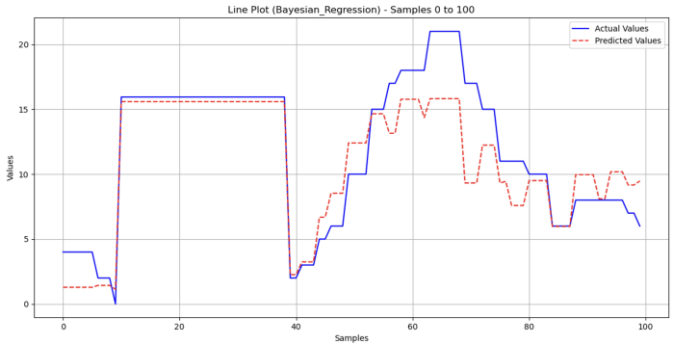


(f) LSTM

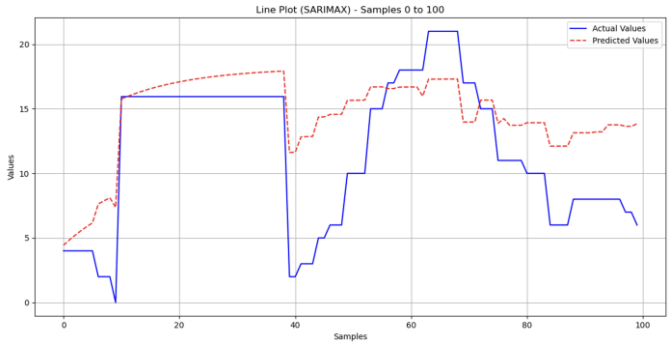


(g) GTT

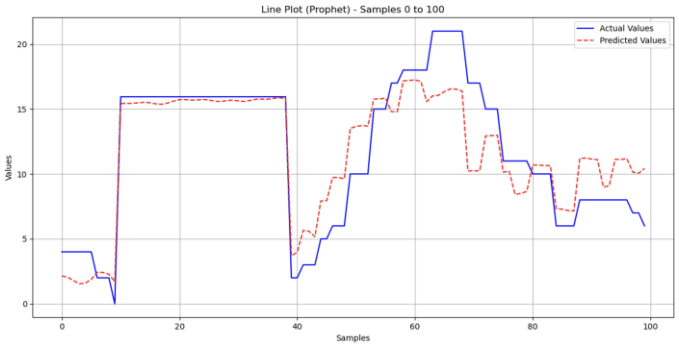
Figure 47. Scatterplot Full imputation Dataset



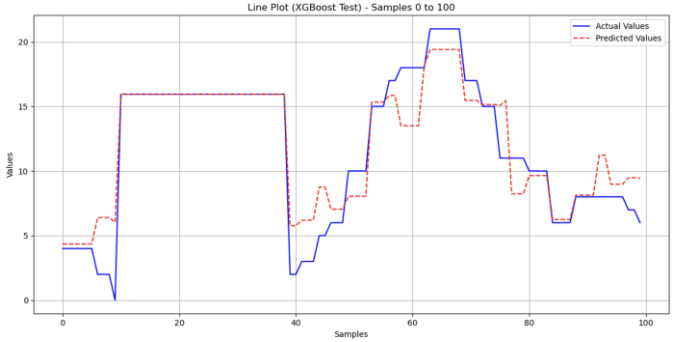
(a) Bayesian Regression



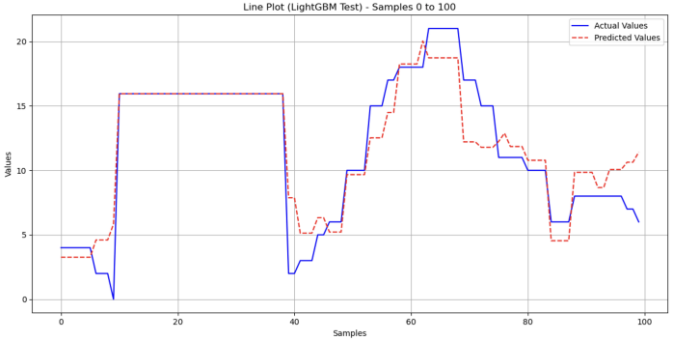
(b) SARIMAX



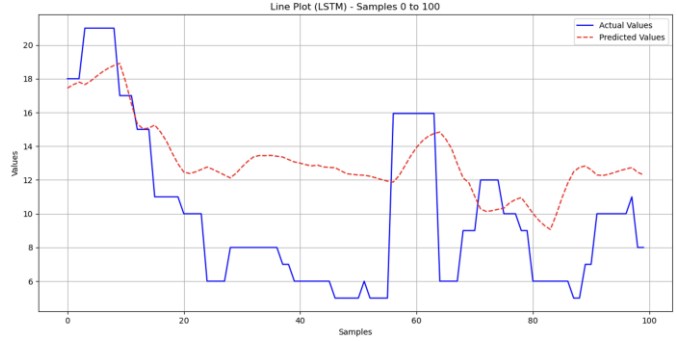
(c) Prophet FB



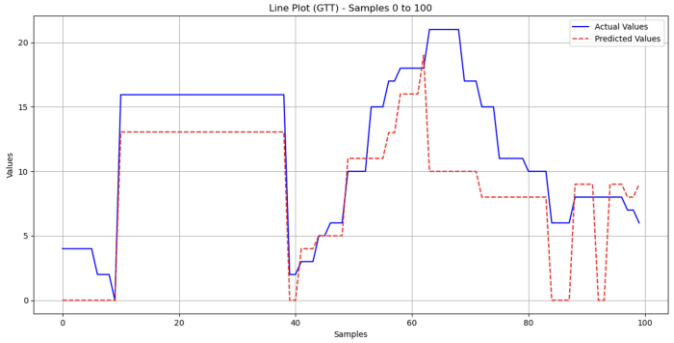
(d) XGBoost



(e) LightGBM

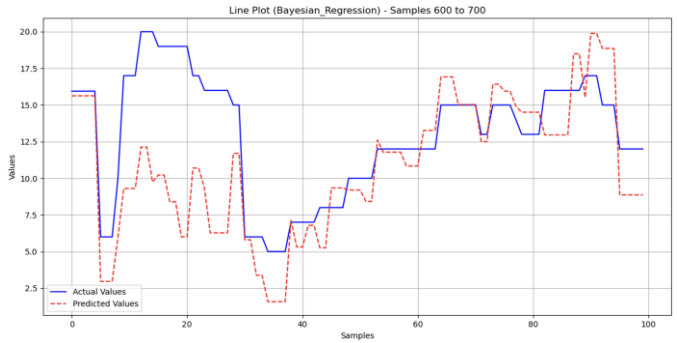


(f) LSTM

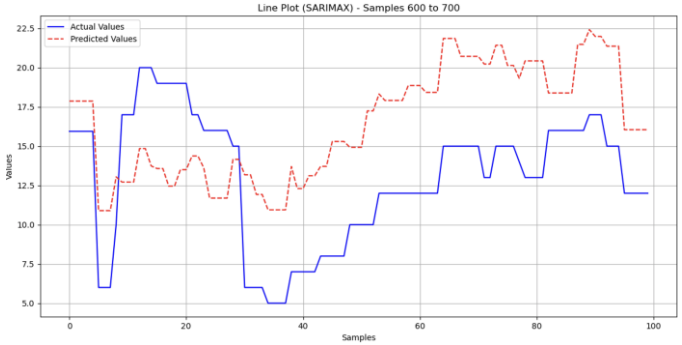


(g) GTT

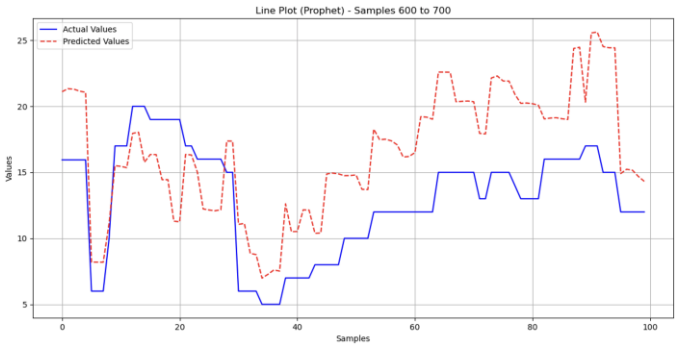
Figure 48. Best datafit graph - LightGBM



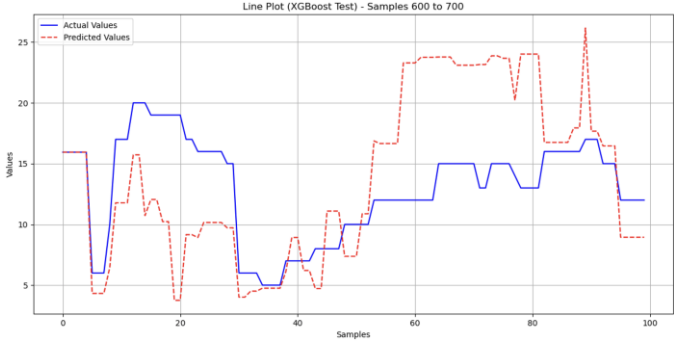
(a) Bayesian Regression



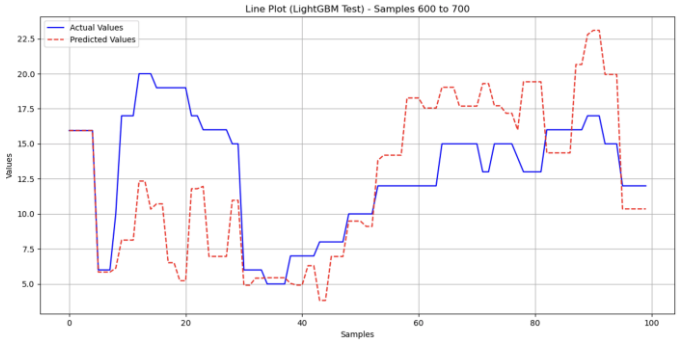
(b) SARIMAX



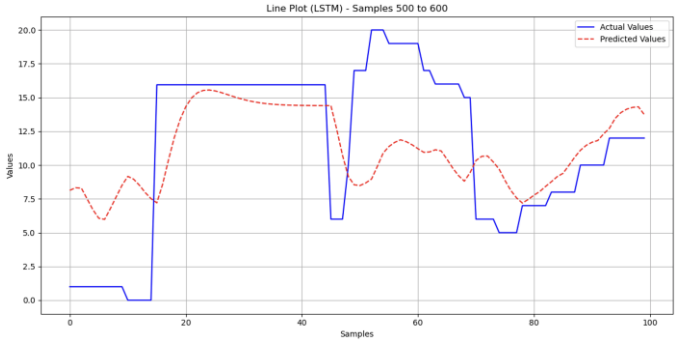
(c) Prophet FB



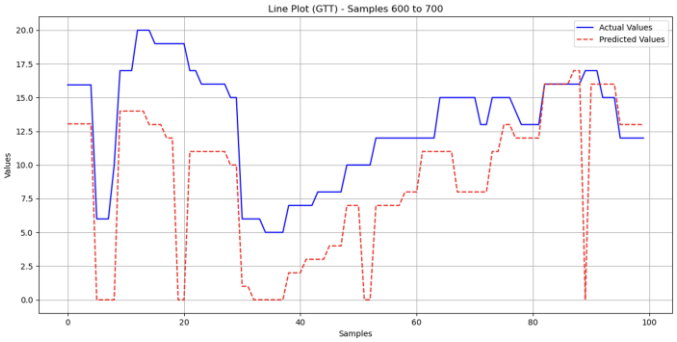
(d) XGBoost



(e) LightGBM

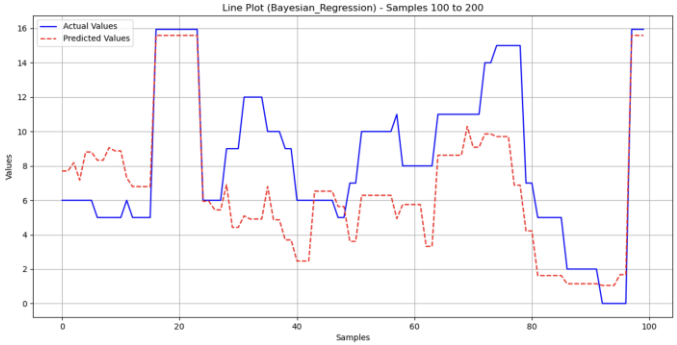


(f) LSTM

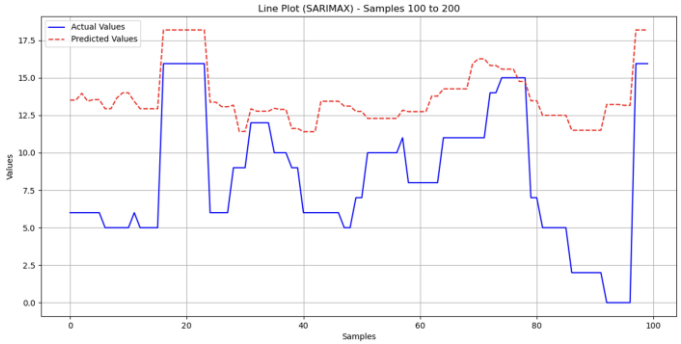


(g) GTT

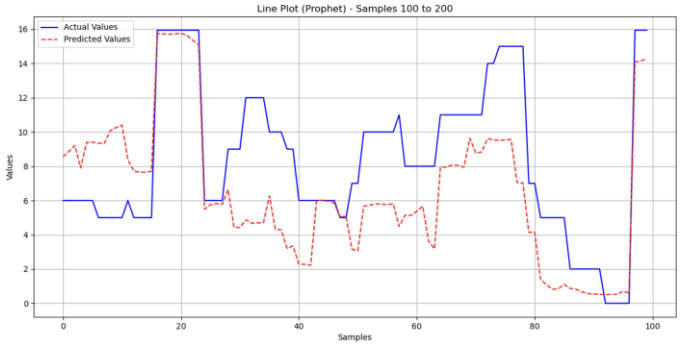
Figure 49. Worst graphs fit - LightGBM



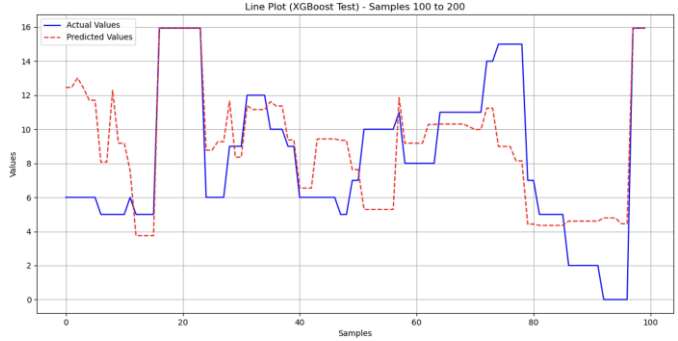
(a) Bayesian Regression



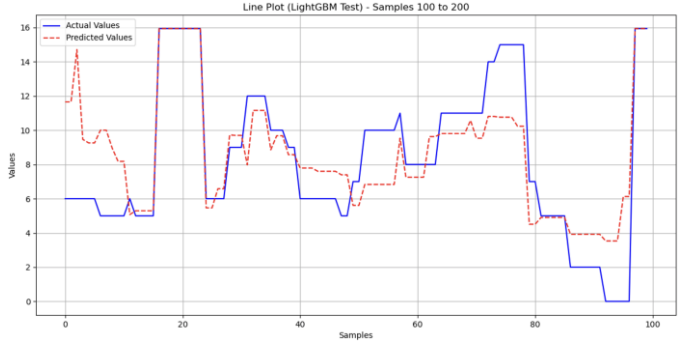
(b) SARIMAX



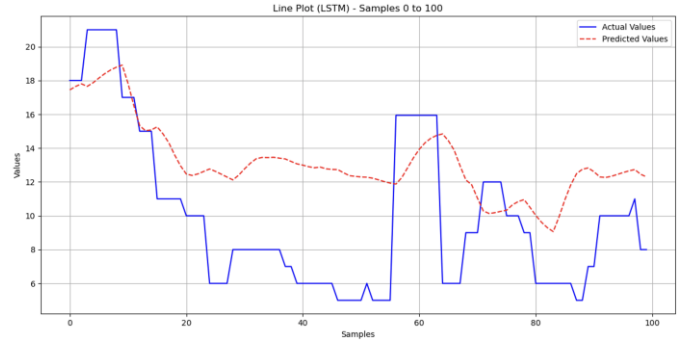
(c) Prophet FB



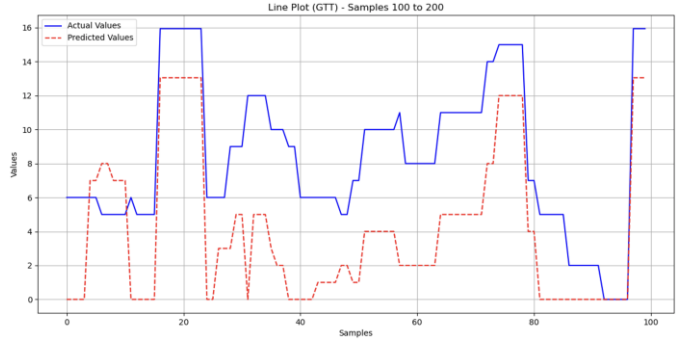
(d) XGBoost



(e) LightGBM



(f) LSTM



(g) GTT

Figure 50. Anomaly behaviour graphs – Full imputation dataset

Train dataset imputation

The application of the algorithms on the final dataset in which only the training dataset was augmented through imputation, yielded results that are highlighted in Table 26. Two immediate observations can be made. Firstly, the accuracy has not significantly improved due to data augmentation, confirming the assumption that the improved accuracy on the dataset, where even the test dataset was augmented, was primarily due to the newly filled rows. Secondly, there is a slight improvement in the performance of both SARIMAX and LSTM compared to the simple dataset, while the performance of XGBoost and LightGBM has slightly degraded. This is understandable as LSTM performs better with larger quantities of data, and SARIMAX relies on seasonality in the data, which can be better captured with a larger dataset available for training. The more surprising finding was the degradation in the performance of the Prophet model, which is known to handle missing data very well. This could be attributed to the introduction of noise through KNN imputation, disrupting the temporal dependencies and seasonality patterns essential for Prophet's accurate modelling. Additionally, the imputed values might have introduced biases that misalign with Prophet's assumptions about the data, leading to suboptimal performance. Furthermore, the imputation process might have overridden Prophet's internal handling mechanisms for missing data, causing it to overfit to the newly filled values and thereby reducing its predictive accuracy on actual future data points.

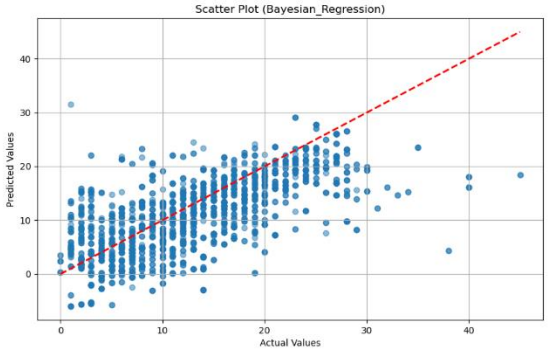
Table 26. Performance evaluation - Train imputation dataset

TRAIN DATASET IMPUTATION				
model	**Accuracy (%)**	**MAE**	**pearson_corr**	**spearman_corr**
SARIMAX	77.90548949	4.6	0.648740117	0.657595298
LSTM	77.13581114	4.8	0.622268793	0.630800628
XGBoost Test	76.48036315	4.93	0.672179543	0.682946371
LightGBM Test	76.3738017	4.51	0.742730389	0.76000808
Bayesian_Regression	74.02283451	4.65	0.656308803	0.674879334
GTT	62.05366626	5.36	0.659796819	0.69221466
Prophet	58.70339633	7.14	0.378479935	0.400060927

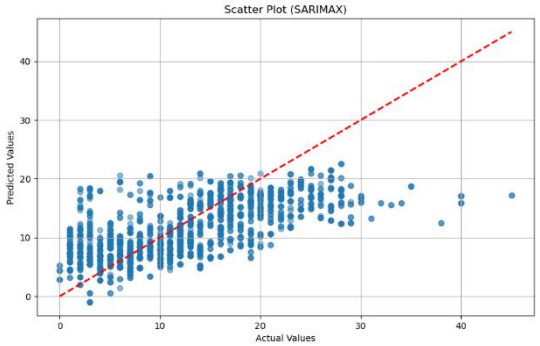
The scatter plots for the models are presented in Figure 51. It is evident from these plots that both Prophet and XGBoost have visibly degraded in performance compared to the simple dataset, while SARIMAX and LSTM have shown slight improvements, becoming more compact and aligned with the reference line. Examining the best fit graphs in Figure 52 , it is clear that SARIMAX has performed the best, displaying a very close fit to the ground truth. LSTM also shows a marked improvement compared to the simple dataset. The Bayesian algorithm has maintained its performance from the simple dataset, whereas XGBoost and LightGBM have clearly experienced a slight degradation in their performance.

In the worst-fit graphs presented in *Figure 53*, Prophet performs particularly poorly, exhibiting severe underestimation compared to reality. This issue is also prevalent in most other models, with Bayesian being the least affected among the poorly performing models.

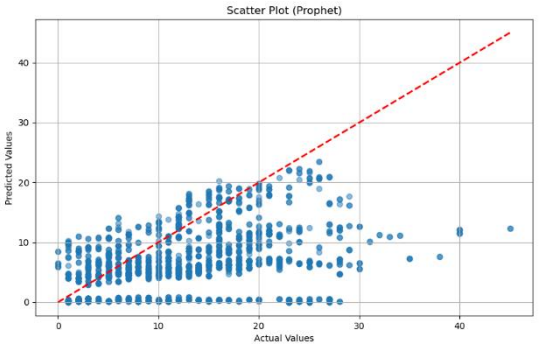
Examining the anomaly graphs in *Figure 54*, there is a discernible trend of overestimation in nearly all models (except Prophet) compared to reality, whereas the GTT dataset demonstrates reasonable tracking. This discrepancy appears to be attributable to the time being close to afternoon school peak hours and the presence of high-density building areas. The models likely lack sufficient data for all possible combinations, such as scenarios where, even in high-density building areas during peak times, crowding is less pronounced if the stop is not proximate to a school. Overall, many of the anomalies observed in the various plots seem to arise from data deficiencies, resulting in a consequent lack of generalisability in the models.



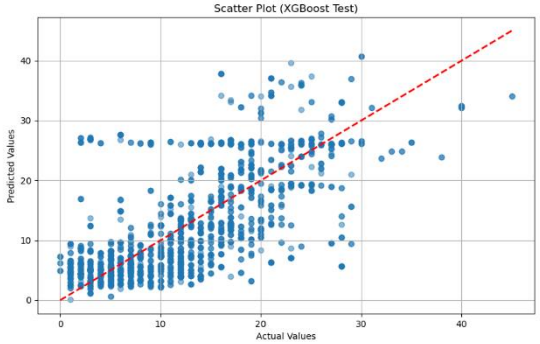
(a) Bayesian Regression



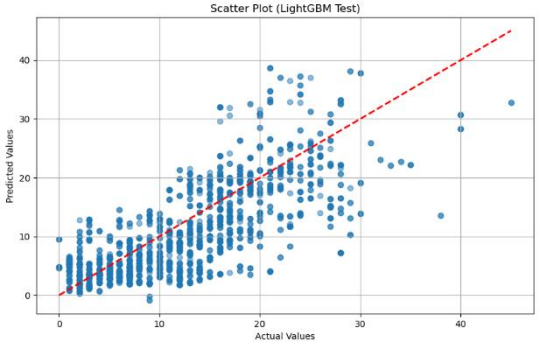
(b) SARIMAX



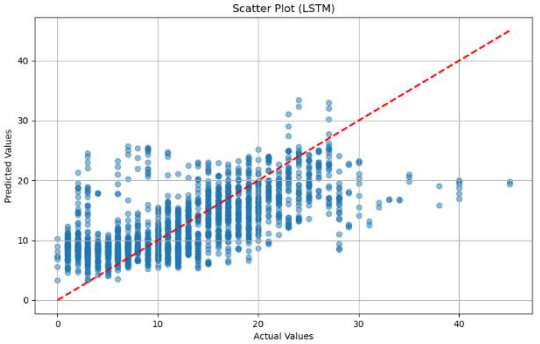
(c) Prophet FB



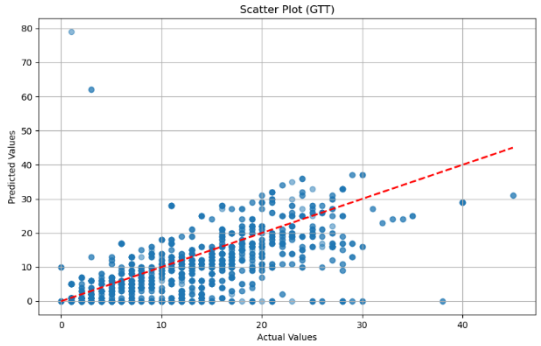
(d) XGBoost



(e) LightGBM

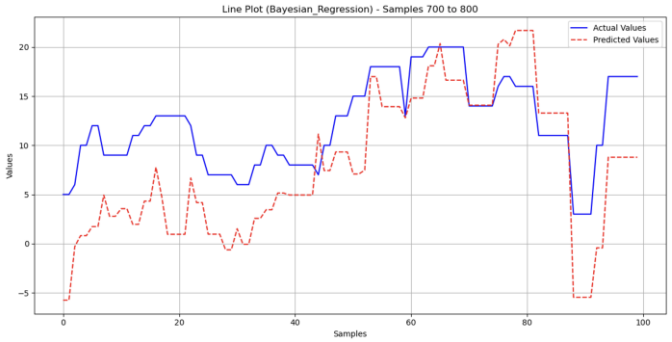


(f) LSTM

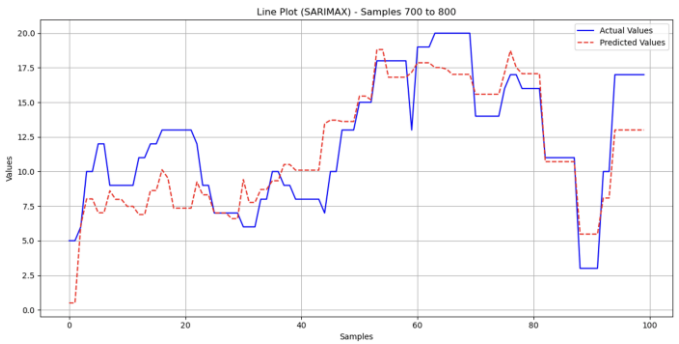


(g) GTT

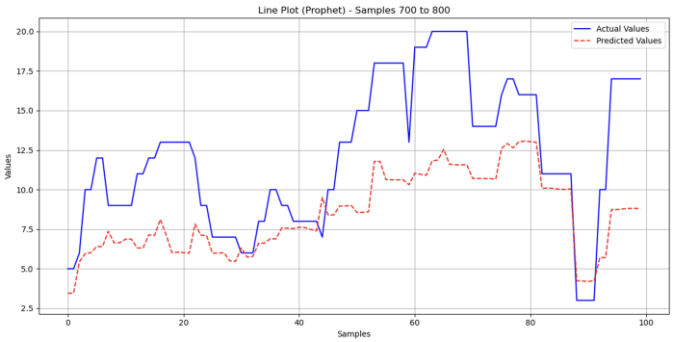
Figure 51. Scatterplot Train imputation Dataset



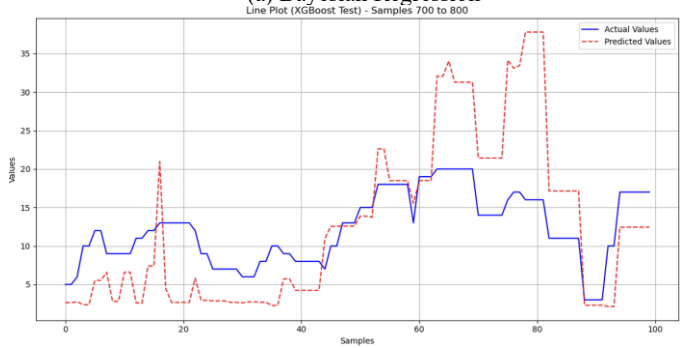
(a) Bayesian Regression



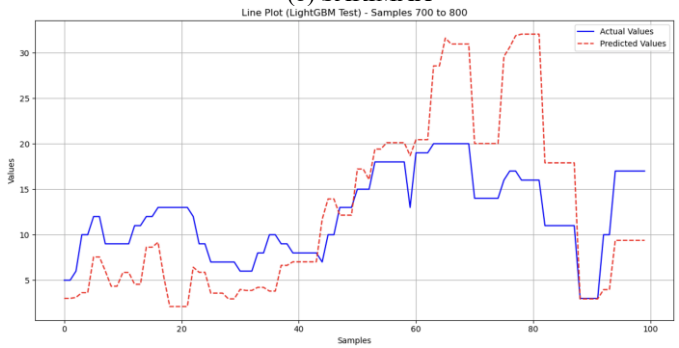
(b) SARIMAX



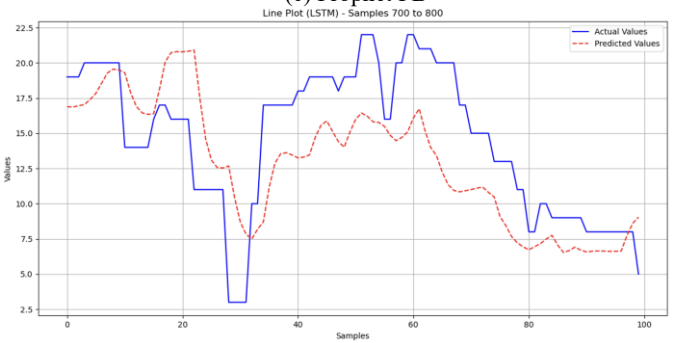
(c) Prophet FB



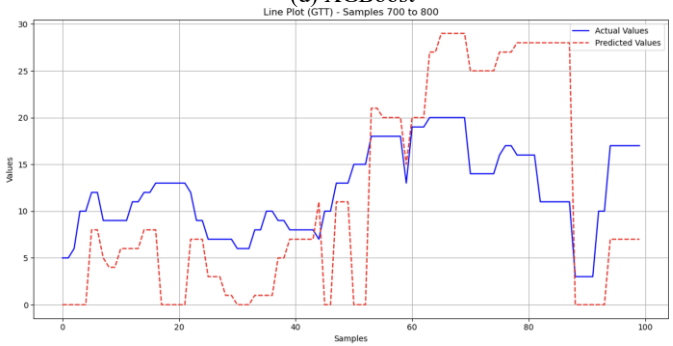
(d) XGBoost



(e) LightGBM

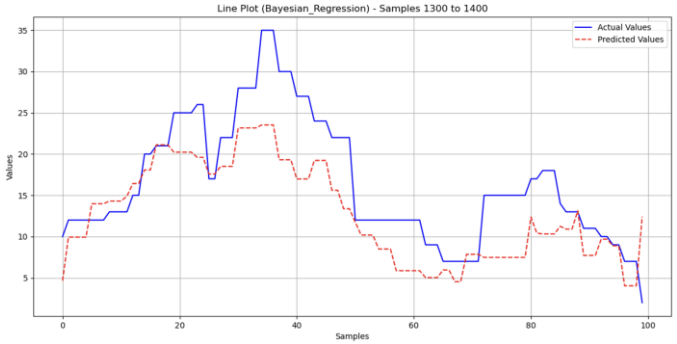


(f) LSTM

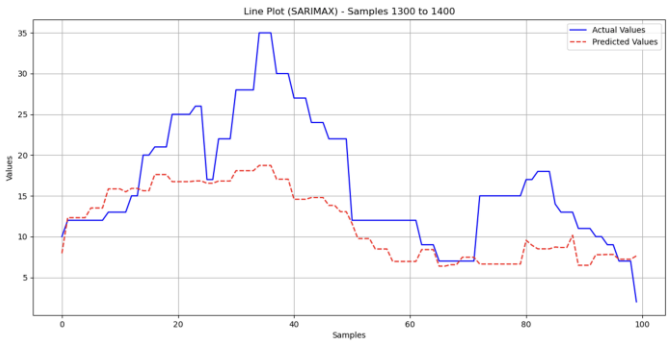


(g) GTT

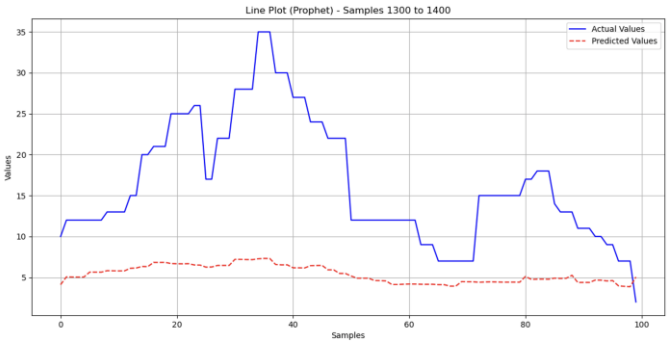
Figure 52. Best datafit graph - SARIMAX



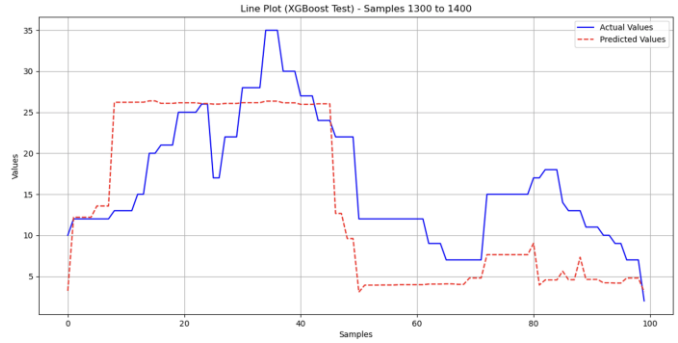
(a) Bayesian Regression



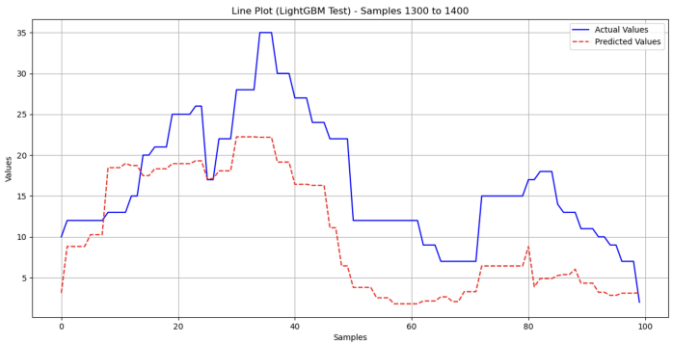
(b) SARIMAX



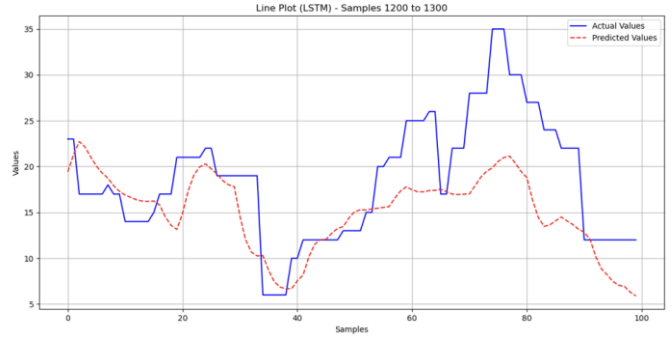
(c) Prophet FB



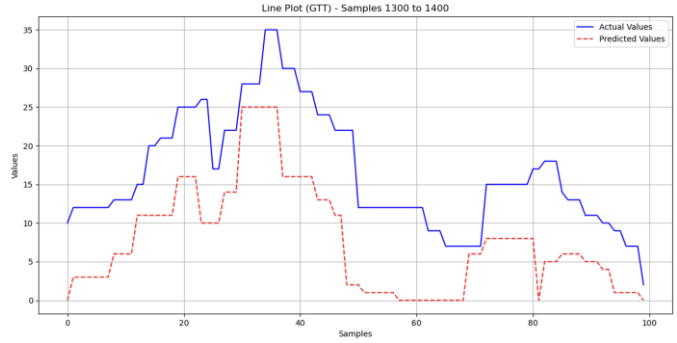
(d) XGBoost



(e) LightGBM

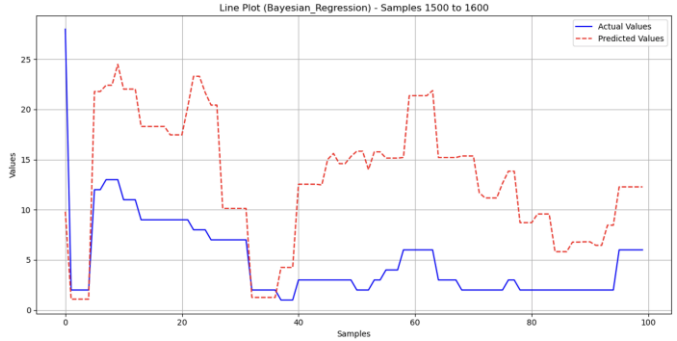


(f) LSTM

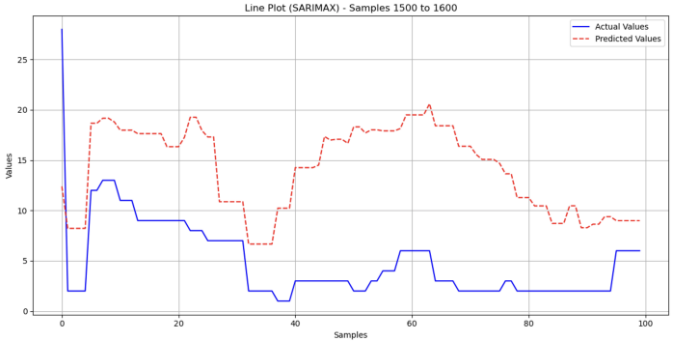


(g) GTT

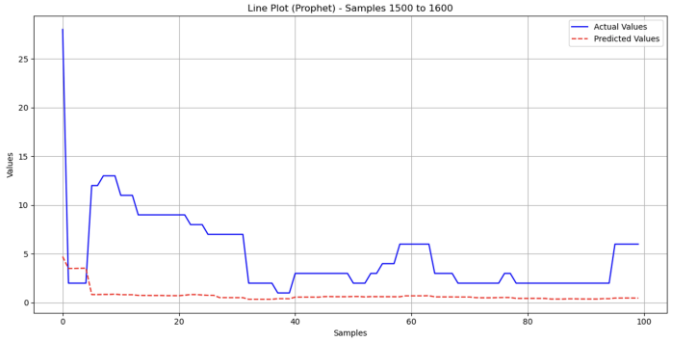
Figure 53. Worst datafit graph - SARIMAX



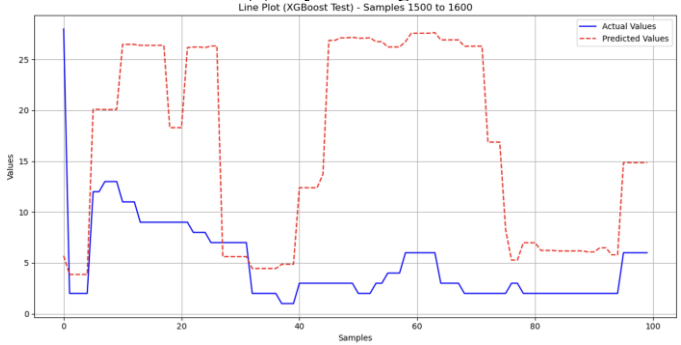
(a) Bayesian Regression



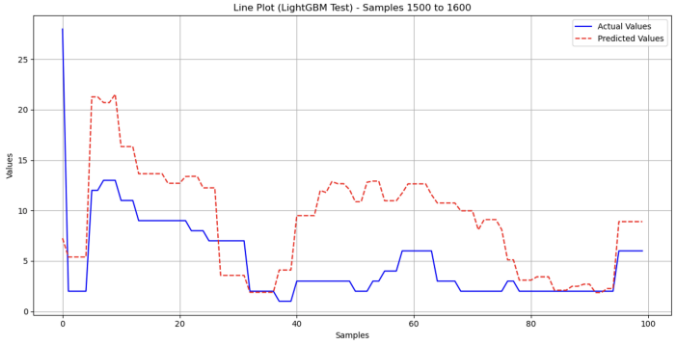
(b) SARIMAX



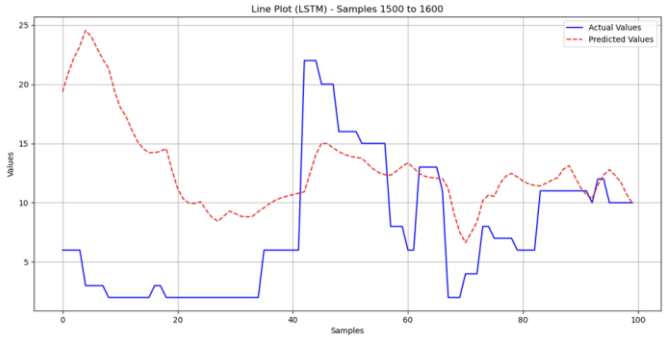
(c) Prophet FB



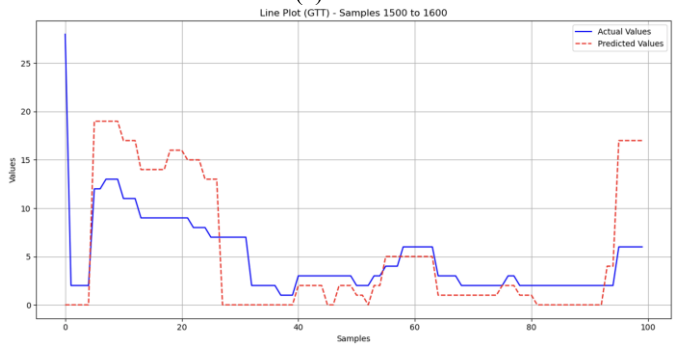
(d) XGBoost



(e) LightGBM



(f) LSTM



(g) GTT

Figure 54. Anomaly behaviour graphs – Train imputation dataset

4.3 Comparative Analysis

After the results of the different models using the accuracy metrics and plots is verified, in the following section, a comparative analysis of the different models is presented. In section 4.3.1, the various factors considered for analysing the models are outlined, followed by the weights assigned to each factor. The final decision matrix is then presented as the concluding element of this comparative analysis.

4.3.1 Factors evaluation

During the training, the algorithms resource usage and used time were collected, in addition to the accuracy metrics. Table 27 shows the accuracy metrics along with the training time and memory used. The accuracy metrics are presented from here on only for the simple dataset as they produced reasonable results across all models and also do not have any bias or noise introduced by the imputation techniques. But for other factors like training time and memory used, the value from the fully imputed dataset is used as this is purely a measure of how long the model takes to fit for the training data and how much memory it uses during this process. The time is presented in seconds and the memory used for the training process in Mebibytes (MiB) which is very close to Megabytes (MB) in the decimal system.

Table 27. Performance factors

Model	training time	Memory used training	Accuracy (%)	MAE	Pearson corr	Spearman corr
Bayesian_Regression	4.027577	3.385417	75.02374501	4.49	0.670021	0.688265
LightGBM	1.984812	3.065104	76.25944678	4.43	0.759806	0.77714
LSTM	146.8811	124.5065	73.61184962	5.27	0.562067	0.56343
Prophet	13.91744	19.39453	70.49138654	5.29	0.583634	0.61082
SARIMAX	886.2459	3652.697	75.50881582	5.24	0.573341	0.60508
XGBoost	2.004041	35.46354	77.66494809	4.55	0.700894	0.72130

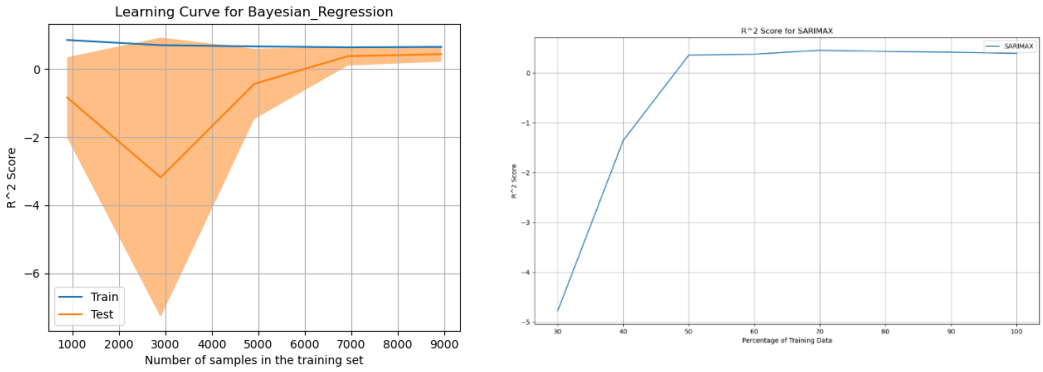
To further evaluate the scalability of the models in terms of time usage and memory usage, some other values are calculated and presented in *Table 28*. The main idea behind the table was the calculation of increase in training time and memory used for 10% increase in data size. However, if one considers only this percentage value, it might appear that LightGBM is not a very scalable model in terms of memory utilisation. Hence, average and maximum values from a set of

training time and memory usage values are also presented. These were obtained by training all the models using different sample dataset sizes on the full imputation dataset.

Table 28 Scalability of models - some numbers

Model	Avg Time Increase per 10% Data Increase (%)	Average Time Usage (seconds)	Highest Time Usage (seconds)	Avg Memory Increase per 10% Data Increase (%)	Average Memory Usage (MiB)	Highest Memory Usage (MiB)
Bayesian Regression	0.223439084	3.996263536	4.288892	14.28571	0.036198	0.390625
LightGBM	0.133748812	2.253846169	2.395175	113.8478	3.852865	9.496094
LSTM	8.425745861	98.50531953	145.0105	7.247089	64.49219	92.16016
Prophet	14.01387943	2.311632442	3.69267	138.6155	3.946615	9.382813
SARIMAX	12.33889262	116.5346515	220.1479	23.03214	405.6693	878.4453
XGBoost	2.469798567	1.718362315	2.069878	17.64317	20.08594	33.16016

Furthermore, learning curves were also generated using the r^2 score which helps evaluating the goodness of fit of a regression model, providing insights into how well the model captures the underlying patterns in the data. An r^2 score of 1 indicates that the regression model could successfully predict the dependent variable, whereas an r^2 score of 0 suggests that the model does not explain any of the variability in the dependent variable. Negative values of r^2 are also possible and indicate that the model performs worse than the mean of the dependent variable (GeeksforGeeks, 2023). Except for the SARIMAX and Prophet models which had library compatibility issues, these scores were calculated using scikit learn library. Despite the efforts put into the customised construction of learning curves for the SARIMAX and Prophet models, it was not possible to obtain a good learning curve from the Prophet algorithm. Figure 55 presents the r^2 score for the statistical models Bayesian Regression (Figure 55a) and SARIMAX (Figure 55b).



(a) (b)

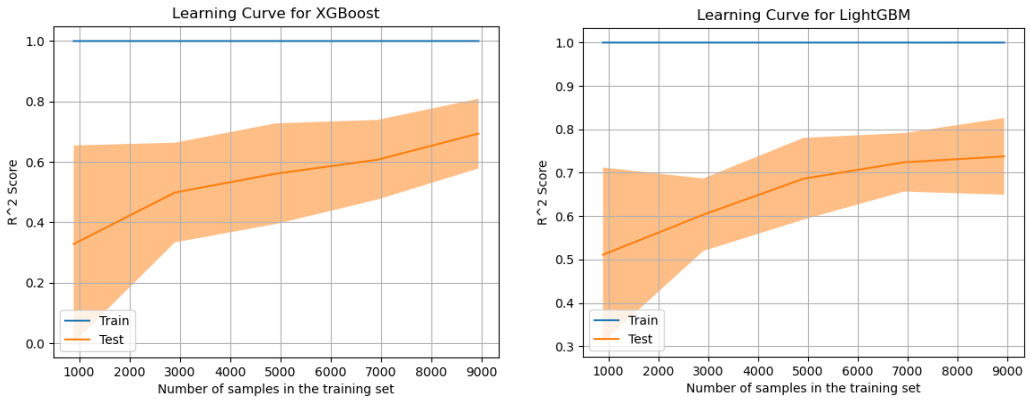
Figure 55. r2 score over sampling size for Statistical models: (a) Bayesian Regression and (b) SARIMAX

For the Bayesian regression, the test data did not follow the consistently increasing trend with smaller sample sizes, indicating that the model initially struggled to explain the variability of the dependent variable. However, with a sample size of 7000, the r^2 value reached 0.2, showing small increments as the data size was increased, eventually reaching 0.35. This suggests that the Bayesian model could potentially improve its performance with a larger dataset. There is also considerable convergence between the curves for the training and test datasets.

For the SARIMAX model, the learning curve was done using custom built approach. The r^2 score improved with an increase in records but flattens at around 0.4 with a sample size of approximately 4500. This shows limited improvement potential with a larger dataset, suggesting that the model's average performance was achieved with the data quantity used in this study. However, performance could be influenced by the quality and structure required for a time-series dataset, especially in terms of continuity and repeated seasonality which was lacking in the dataset.

On the other hand, the ML models show a better performance than the statistical models in terms of learning curves, as shown in Figure 56. Both models exhibit high training r^2 scores, consistently close to 1.0, indicating excellent fit on the training data. The test r^2 scores for both models increase steadily with the number of training samples, with LightGBM's test R^2 score reaching approximately 0.7 and XGBoost's reaching around 0.75 as the sample size approaches 9000. This trend suggests that adding more data improves the models' performance on the test set, although the rate of improvement diminishes slightly as the sample size grows

especially for LightGBM. A significant gap for small sample sizes between the training and test scores for both models indicates potential overfitting, although this gap narrows with increasing sample sizes. A key difference between the two plots is that XGBoost achieves a slightly higher test R^2 score compared to LightGBM at larger sample sizes and the curve has not flattened as much as LightGBM, suggesting that XGBoost may generalise marginally better in this scenario.



(a) (b)

Figure 56. r2 score over sampling size for ML models: (a) XGBoost and (b) LightGBM

The learning curve for LSTM, as presented in Figure 57, reveals a unique and somewhat atypical pattern. Initially, the training R^2 score is notably low, indicating that the model struggles to learn effectively from a limited dataset, a common issue with high-capacity models like LSTMs. As the number of training samples increases, there is a marked improvement in both training and test scores, suggesting that the model benefits significantly from a larger dataset. This improvement reflects enhanced generalisation capabilities and reduced overfitting, evidenced by the narrowing gap between training and test scores.

The initial low scores and high variance indicate the model's difficulty in learning from small datasets. However, the significant increase in performance with more data suggests that the model will continue to benefit from further data growth. This trend is supported by decreasing variance across cross-validation folds, pointing to more reliable and consistent performance as the dataset grows. Therefore, it is advisable to gather additional training data to enhance the model's performance further. Monitoring the learning curves as more data are added will be crucial to ensure the LSTM model achieves optimal performance, characterised by improved accuracy and reduced overfitting.

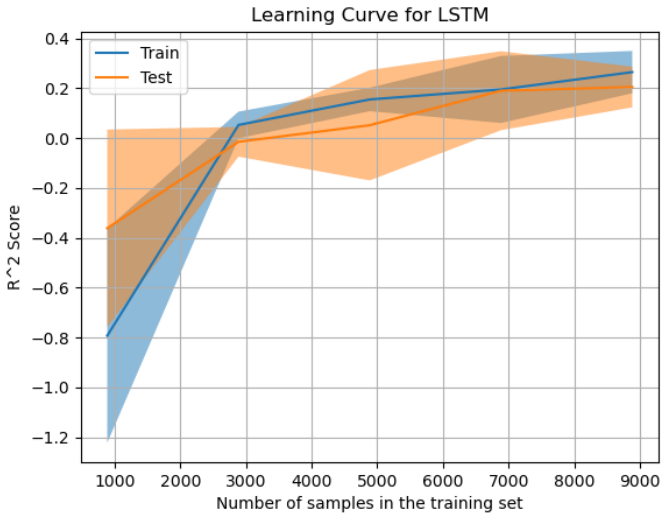


Figure 57. r2 score over sampling size for LSTM model

After observing the learning curves for the r^2 scores, the model's data quantity requirements were defined, as shown in Table 29. The model that requires the most data to improve its performance, similar to our dataset, is LSTM. In contrast, SARIMAX may not require as much data to achieve good performance.

Table 29. Data quantity requirements

Model	Sample size	R2 score	Data quantity needed
Bayesian_Regression	7000	0.35	2
SARIMAX	4400	0.4	1
Prophet	-	-	4
XGBoost	>9000	0.5	4
LightGBM	>9000	0.73	3
LSTM	>9000	0.21	5

The factor matrix used for the decision matrix is presented in *Table 30*.

Table 30. Factors for Decision Matrix

Model	Average Time Usage (seconds)	Average Memory Usage (MiB)	Data quantity needed	Accuracy (%)	MAE	spearman_corr
Bayesian_Regression	3.996264	0.036198	2	75.02375	4.49	0.688265
LightGBM	2.253846	3.852865	3	76.25945	4.43	0.777144
LSTM	98.50532	64.49219	5	75.96314	4.8	0.645445
Prophet	2.311632	3.946615	4	70.49139	5.29	0.610823
SARIMAX	116.5347	405.6693	1	75.50882	5.24	0.605085
XGBoost	1.718362	20.08594	4	77.66495	4.55	0.721302

4.3.2 Weights calculation

In this section, the results of the Paired Comparison Analysis used to calculate the weights of the Decision matrix are shown. First, each factor was assigned a letter, as shown in Table 31.

Table 31. Letter assigned to factors

Factor	Letter
Average Time Usage (seconds)	A
Average Memory Usage (MiB)	B
Data quantity needed	C
Accuracy (%)	D
MAE	E
spearman_corr	F

Then, these factors were compared in a factor-by-factor matrix, in which the letter representing the more important factor between each row and column pair was written and a value of importance was assigned, from 0 (same importance) to

3 (one much more importance). The guiding question for this step was: "How important is the winning factor compared to the losing factor?". This evaluation is reported on Figure 58, then the letter values were added up and converted into percentages of the total. These calculations yield the totals shown in Table 32

LETTER		Average Time Usage (seconds)	Average Memory Usage (MiB)	Data quantity needed	Accuracy (%)	MAE	spearman_corr
LETTER		A	B	C	D	E	F
Average Time Usage (seconds)	A		A1	C3	D3	E2	F1
Average Memory Usage (MiB)	B			C3	D3	E2	F2
Data quantity needed	C				C1	C1	C2
Accuracy (%)	D					D0	D3
MAE	E						E2
spearman_corr	F						

Figure 58. Pair comparison matrix

The comparison was made considering the needs of a transport company, based on the experience achieved collaborating with GTT during this study. The most important factor is data quantity, followed closely by accuracy, with a difference of 3%. Although high accuracy is typically the most critical variable in model performance, data collection poses a significant challenge in the transport sector, particularly in public transport. Thus, data quantity was given slightly more weight in the decision matrix.

Table 32. Weight definition

Factor	Letter	Letter count	Percentage (%)
Average Time Usage (seconds)	**A**	1	3.333333
Average Memory Usage (MiB)	**B**	1	3.333333
Data quantity needed	**C**	10	33.33333
Accuracy (%)	**D**	9	30
MAE	**E**	6	20
spearman_corr	**F**	3	10

On the other hand, during the comparison shown in Figure 58, it was observed that no value was initially assigned to factor B. However, for the purposes of

defining weights, a value was assigned to factor B to ensure it had a non-zero weight percentage in the decision matrix. In terms of importance, it is fair to say that average time usage (factor A) and average memory usage (factor B) are equally important. Therefore, the value assigned to factor B is the same as that assigned to factor A.

4.3.3 Decision matrix

After defining the factors and their decision weights, reported respectively in *Table 30* and *Table 32*, normalisation was done to ensure the same scale of evaluation for all factors. Moreover, an inverted normalisation was applied to the factors where a lower value indicates better performance, as shown in *Table 33*. The normalized matrix is reported in *Table 34*

Table 33. Inverted Normalization for factors

Factor	Value level for good performance	Inverted normalization
Average Time Usage (seconds)	low	Yes
Average Memory Usage (MiB)	low	Yes
Data quantity needed	low	Yes
Accuracy (%)	high	No
MAE	low	Yes
spearman_corr	high	No

Table 34. Normalized decision matrix

Model	Average Time Usage (seconds)	Average Memory Usage (MiB)	Data quantity needed	Accuracy (%)	MAE	spearman_corr
Bayesian_Regression	0.98016	1	0.75	0.631814	0.930233	0.483442
LightGBM	0.995336	0.990591	0.5	0.804072	1	1
LSTM	0.157028	0.841098	0	0.762766	0.569767	0.234574
Prophet	0.994833	0.99036	0.25	0	0	0.033352
SARIMAX	0	0	1	0.699434	0.05814	0
XGBoost	1	0.950572	0.25	1	0.860465	0.675447

The normalised matrix was then multiplied by each factor's weight, and the resulting values were summed for each model. The model with the highest total score was identified as the best. According to the decision matrix reported in *Figure 59*, and considering the dataset and other performance factors in this study, the best model is LightGBM, followed by Bayesian regression and XGBoost.

Weights	3.333333	3.333333	33.33333	30	20	10	
Letter	A	B	C	D	E	F	
Model	Average Time Usage (seconds)	Average Memory Usage (MiB)	Data quantity needed	Accuracy (%)	MAE	spearman_corr	TOTAL
Bayesian_R	3.267202	3.333333	25	18.95443	18.60465	4.834422	73.99404
LightGBM	3.317787	3.301969	16.66667	24.12216	20	10	77.40858
LSTM	0.523425	2.803659	0	22.88298	11.39535	2.345736	39.95115
Prophet	3.31611	3.301199	8.333333	0	0	0.33352	15.28416
SARIMAX	0	0	33.33333	20.98301	1.162791	0	55.47913
XGBoost	3.333333	3.168572	8.333333	30	17.2093	6.754473	68.79901

Figure 59. Final decision matrix

Chapter 5

Discussion and Conclusions

The transport sector employs various methods for data collection in the public transport, including different types of automated passenger counting (APC) and automated fare collection (AFC) systems, as well as traditional methods like travel surveys. Efforts in the literature have focused on enhancing univariate collected data for prediction and categorisation problems to improve the transport system, such as optimising demand-responsive bus services within the traditional FRFS framework. Additional efforts have been made on fusing these datasets with external variables such as territorial variables, meteorological, and environmental data, which can provide behavioural and decisional insights about users. Indeed, the use of external variables has been confirmed to improve prediction accuracy.

Multiple methods have been used for data fusion, including statistical methods, ML, and DL algorithms, chosen based on analysis type, complexity, and interpretability. Statistical models are capable to infer relationships between variables and make predictions, while ML leverages data-driven methods to enhance performance, and DL uses complex neural networks to learn patterns. DL models learn high-level features, eliminating the extensive feature extraction needed by statistical and some ML models. Despite DL's promise, it faces challenges like interpretability, higher data requirements, and significant computational demands. Literature suggests DL excels with large datasets, whereas traditional statistical models and ML are better with smaller datasets. Hybrid methods combining statistical and ML models have also been tested, structured to estimate missing features and estimate passenger load.

Although each method presented in the literature claims to be able to predict passenger load through data fusion with good accuracy and low error, determining the best method for a specific dataset remains a challenge because there are no studies comparing these models with real operational data collected manually or using the same database, also considering factors such as dataset size and computing power. Moreover, despite the extensive application of both artificial intelligence (AI) models and statistical models in transport analytics, the literature provides inconclusive evidence regarding the superiority of one approach over the other in terms of results. Additionally, models from other fields with strong potential for achieving high accuracy and performance in transport applications have yet to be explored, and the influence of dataset size on the choice of modelling technique remains underexamined.

In light of these challenges, this research aimed to conduct a thorough empirical examination of passenger count estimation models developed using both AI and traditional statistical modelling techniques using manually collected real operation data as a ground truth. The study intended not only to compare their performance, accuracy, and data requirements but also to explore the efficacy of data fusion methodologies in capturing mobility patterns, the scalability and applicability of such methodologies and its representation of reality.

To achieve these objectives, a nine-month measurement campaign allowed to obtain 57 days of ground truth data collected from buses during operation times in Turin. Data fusion was then performed with 6 different algorithms: Bayesian regression, SARIMAX, FB prophet, XGBoost, LightGBM and LSTM. These algorithms used a single database containing information from two APC systems (camera-based and Wi-Fi-based) along with environmental data, including census data, POIs, meteorological data, and pollution data. The performance of these models was compared using the manually collected data as a benchmark.

Moreover, a pilot study was conducted to understand potential data collection challenges and ensure high-quality data, in which a low-cost APC system was built and tested on field operations. Finally, a comparative analysis was conducted using the decision matrix method, a MCDA technique, to compare AI and statistical models based on performance, data requirements, and scalability, ultimately defining the best overall model.

According to the decision matrix considering the dataset and performance factors, the best-fitted model for our context was the LightGBM, followed by

Bayesian regression and XGBoost. These findings are supported by the performance results of the models in our three experiments. In the first experiment with the simple dataset, XGBoost achieved the highest accuracy at 78%, while LightGBM had the lowest error of 4.43 and the best correlation. While, in the experiment with full imputation dataset, LightGBM was the best-performing model in terms of accuracy, error, and correlation, achieving an accuracy of 89.6%. In the training dataset of the imputation experiment, SARIMAX achieved the highest accuracy of 77.9%, but LightGBM again recorded the lowest error and highest correlation.

Bayesian regression was the second-best fit for our study, consistently achieving one of the three lowest errors in all experiments. Additionally, our results suggest that it does not require a large dataset to produce good results, offering a good compromise between performance and data quantity.

An interesting point to note is that, only for SARIMAX and Prophet, the Pearson correlation improved considerably from the simple dataset experiment to the full imputation dataset experiment, increasing, respectively, from 0.573 and 0.583 to 0.700 and 0.701. This fact suggests that data augmentation has helped these two algorithms that rely the most on seasonality, benefitting from having continuous time series data with some seasonality, even if a significant portion was augmented using imputation techniques.

On the other hand, it is interesting observing that two of the most advanced time series algorithms used in this study, LSTM and Prophet, proved to be the least effective solutions for the problem at hand. This highlights the importance of analysing the data and selecting the algorithm based on the data's characteristics, such as size, distribution, seasonality, and other patterns. A more modern algorithm cannot produce great results if it does not match the dataset requirements – both in terms of size (as with LSTM) and specific data features (such as seasonality for Prophet). This underscores the importance of collecting a large quantity of high-quality data with good distribution.

These results offer valuable insights for both research and practice, particularly in terms of accurate representation of real-world scenarios. For research, the comparative analysis of these models highlights the importance of considering the characteristics of the dataset and real operating conditions when developing and selecting predictive models. This contributes to the ongoing refinement of the

representation of reality in modelling, ensuring that predictions are more closely aligned with actual outcomes.

From a practical point of view, transport companies can use these results to better understand which models are most applicable to their specific and operating context, particularly when facing limitations in data size or computational resources. LightGBM, for example, has proven to be a robust choice across various datasets, offering both accuracy and efficiency. The knowledge gained can be directly applied to improve passenger load estimates, optimise resource allocation, and improve operational decision-making. Transport companies can apply these results to fine-tune their predictive models, ensuring to choose the algorithm best suited to the specific characteristics of their data, thus improving the accuracy of passenger load predictions in real operations.

However, there are constraints in practically implementing these results. A major challenge is the availability and quality of data. Although the study shows that algorithms like LightGBM can perform well even with scarce data, the results also emphasise the need for a well-distributed, high-quality dataset to achieve optimal results. Furthermore, the computing power required to run more advanced models may be a limiting factor for some transport companies, particularly when scaling these solutions across different regions or systems. In addition, integrating these models with existing infrastructure requires significant effort and additional resources that transport companies may be reluctant to provide. Addressing these constraints is crucial to ensure that the models not only perform well in controlled environments but also offer practical benefits in real-world operational contexts.

From the results, we observed that GTT APC estimation underperforms compared to the tested algorithms, confirming that commercial APC may not perform as well as claimed (Pronello and Garzón Ruiz, 2023; Pronello *et al.*, 2024). In addition, these findings are consistent with the notion that data fusion techniques provide a real improvement when predicting vehicle load (Hoang *et al.*, 2016; Shiao *et al.*, 2017; Bachir *et al.*, 2019; Gallo *et al.*, 2023; Mei *et al.*, 2023). By testing multiple data fusion models with the same database, our study provides insight into determining the model that could potentially best fit similar transport problems. To the best of author's knowledge, this is the first study of its kind, in which different data fusion models are tested and compared using real operation data as a ground truth and considering additional performance metrics such as data size, computational power, and time usage. Additionally, the results from the decision

matrix could serve as a guideline for deciding which model to use in similar estimation problems within the transport field.

Moreover, the results of this study challenge the literature (Nguyen *et al.*, 2018; F. Lu *et al.*, 2021; Sari Aslam *et al.*, 2021; Lu *et al.*, 2023; Yousuf *et al.*, 2024; Zaraket *et al.*, 2024) regarding the expectations for high end models like LSTM and prophet because the results do not show promising outcomes for these algorithms, suggesting that their effectiveness may be overestimated in certain contexts. This finding indicates that advanced algorithms may not always outperform simpler models, especially when the dataset lacks specific characteristics such as sufficient size or appropriate seasonal patterns. It emphasizes the necessity of carefully considering the nature of the data and the suitability of the model to achieve optimal performance, rather than relying solely on the perceived superiority of more complex algorithms.

Despite efforts to reduce any margin of error for proper comparison and performance assessment, the lack of a large, merged dataset could have impacted the models and, more importantly, their performance, which is a limitation of this study. Although 57 days of ground truth data were collected, GTT provided only 14 days of valid data for the study, severely limiting the amount of data available for studying the data fusion approaches. This limitation necessitated data augmentation and imputation, as the dataset was not sufficiently large.

Similarly, the use of time-series models in predicting vehicle load is promising and should be explored with a larger dataset that can properly capture seasonality. For this reason, future research should focus on refining data collection and cleaning methodologies to enhance model performance, as well as exploring time-series models for vehicle load estimation and investigating more data fusion algorithms used in different fields.

Furthermore, it would be valuable to understand how these results can be effectively used by the transport company to improve service. Future work could evaluate the potential implementation of more flexible and customised services (Filippi *et al.*, 2023), such as hybrid models combining customised buses with traditional FRFS services, to address the evolving challenges of urban mobility. By optimising passenger load estimation, services can be improved through schedule adjustments and real-time monitoring, enabling a more demand-responsive approach.

This exploration of the challenges and opportunities in using data fusion and machine learning for enhancing transport services suggests significant potential for understanding mobility patterns through advanced data analytics and supporting it through flexible transport services.

In conclusion, the results obtained provide a reasonable starting point and demonstrate the feasibility of using ensemble machine learning algorithms for data fusion in the transport field. However, further research and development are required, particularly in data collection and cleaning, to achieve better results through the use of higher-quality individual datasets. While only a subset of machine learning algorithms was tested as a proof of concept, future studies should include a more comprehensive range of algorithms to produce better results. Ensemble algorithms, in particular, hold promise in the field of data fusion.

References

Ahmad, M., Ahmed, I. and Adnan, A. (2019a) 'Overhead View Person Detection Using YOLO', in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0627–0633. Available at: https://doi.org/10.1109/UEMCON47517.2019.8992980.

Ahmad, Misbah *et al.* (2019b) 'A Deep Neural Network Approach for Top View People Detection and Counting', in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 1082–1088. Available at: https://doi.org/10.1109/UEMCON47517.2019.8993109.

Amato, F. *et al.* (2022) 'Bus Passenger Load Prediction: Challenges from an Industrial Experience', in F. Karimipour and S. Storandt (eds) *Web and Wireless Geographical Information Systems*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 93–107. Available at: https://doi.org/10.1007/978-3-031-06245-2_9.

Amirabadi, M.A., Kahaei, M.H. and Nezamalhosseini, S.A. (2020) 'Novel suboptimal approaches for hyperparameter tuning of deep neural network [under the shelf of optical communication]', *Physical Communication*, 41, p. 101057. Available at: https://doi.org/10.1016/j.phycom.2020.101057.

AMP (2013) *IMQ 2013: Indagine sulla Mobilità delle Persone e sulla Qualità dei Trasporti.*, p. 32. Available at: https://mtm.torino.it/it/dati-statistiche/indagini/.

Anand, G. and Kumawat, A.K. (2021) 'Object detection and position tracking in real time using Raspberry Pi', *Materials Today: Proceedings*, 47, pp. 3221–3226. Available at: https://doi.org/10.1016/j.matpr.2021.06.437.

Anda, C., Erath, A. and Fourie, P.J. (2017) 'Transport modelling in the age of big data', *International Journal of Urban Sciences*, 21(1), pp. 19–42.

Andueza, A. *et al.* (2023) 'Using the Statistical Machine Learning Models ARIMA and SARIMA to Measure the Impact of Covid-19 on Official Provincial Sales of Cigarettes in Spain', *International Journal of Interactive Multimedia and Artificial Intelligence*, IP, pp. 1–15. Available at: https://doi.org/10.9781/ijimai.2023.02.010.

Arpa Piemonte (2024) *L'Agenzia - Arpa Piemonte*, *L'Agenzia*. Available at: https://www.arpa.piemonte.it/chi-siamo/lagenzia (Accessed: 15 March 2024).

Asti Servizi Pubblici S.p.A. (2021) *Carta della Mobilità*. Asti, Italy. Available at: https://www.asp.asti.it/wp-content/uploads/2021/09/All6_CartadellaMobilita-rev-14_09_2021.pdf (Accessed: 25 July 2023).

Azienda Trasporti Milanesi (2023) *Carta della Mobilità ATM 2023*. Milan, Italy. Available at: https://www.atm.it/it/IlGruppo/ChiSiamo/Pagine/CartaMobilita.aspx (Accessed: 22 August 2023).

Bachir, D. *et al.* (2019) 'Inferring dynamic origin-destination flows by transport mode using mobile phone data', *Transportation Research Part C: Emerging Technologies*, 101, pp. 254–275. Available at: https://doi.org/10.1016/j.trc.2019.02.013.

Baltes, M.R. and Rey, J. (1999) 'The "Ins and Outs" of APCs: An Overview of Automatic Passenger Counters', *Journal of Public Transportation*, 2(2), pp. 47–64. Available at: https://doi.org/10.5038/2375-0901.2.2.3.

Bass, L., Clements, P.C. and Kazman, R. (1997) *Software Architecture in Practice*. Addison-Wesley Professional.

van der Bijl, B. *et al.* (2022) 'A Comparison of Approaches for the Time Series Forecasting of Motorway Traffic Flow Rate at Hourly and Daily Aggregation Levels', *Procedia Computer Science*, 201, pp. 213–222. Available at: https://doi.org/10.1016/j.procs.2022.03.030.

Biletska, K. *et al.* (2009) 'Multisensor data fusion for OD matrix estimation', in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3024–3029. Available at: https://doi.org/10.1109/ICSMC.2009.5345929.

Boeri, A., Longo, D. and Palma, M. (2021) 'CLIMATE-NEUTRAL AND SMART CITIES: A EUROPEAN POLICIES' OVERVIEW', *WIT Transactions on Ecology and the Environment*, 253, pp. 3–14. Available at: https://doi.org/10.2495/sc210011.

Boström, H. *et al.* (2007) 'On the Definition of Information Fusion as a Field of Research', *IKI Technical Reports*, (June 2014), pp. 1–8.

Boulkaboul, S. and Djenouri, D. (2020) 'DFIOT: Data Fusion for Internet of Things', *Journal of Network and Systems Management*, 28(4), pp. 1136–1160. Available at: https://doi.org/10.1007/s10922-020-09519-y.

Box, G.E.P. *et al.* (2015) *Time Series Analysis: Forecasting and Control*. 5th edn. John Wiley & Sons (Wiley Series in Probability and Statistics).

Brlek, P. *et al.* (2022) 'Application of fleet management in intelligent transport systems', in *2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*. *2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, pp. 1–4. Available at: https://doi.org/10.1109/ICEST55168.2022.9828676.

Brownlee, J. (2016) *XGBoost With Python: Gradient Boosted Trees with XGBoost and scikit-learn*. 10th edn. Machine Learning Mastery. Available at: https://books.google.it/books?id=HgmqDwAAQBAJ.

Butler, L., Yigitcanlar, T. and Paz, A. (2020) 'Smart Urban Mobility Innovations: A Comprehensive Review and Evaluation', *IEEE Access*, 8, pp. 196034–196049. Available at: https://doi.org/10.1109/ACCESS.2020.3034596.

Cai, J. *et al.* (2018) 'Feature selection in machine learning: A new perspective', *Neurocomputing*, 300, pp. 70–79. Available at: https://doi.org/10.1016/j.neucom.2017.11.077.

Castanedo, F. (2013) 'A review of data fusion techniques', *The Scientific World Journal*, 2013. Available at: https://doi.org/10.1155/2013/704504.

Cecaj, A. *et al.* (2021) 'Sensing and Forecasting Crowd Distribution in Smart Cities: Potentials and Approaches', *IoT*, 2(1), pp. 33–49. Available at: https://doi.org/10.3390/iot2010003.

Cepeliauskaite, G. *et al.* (2021) 'Smart-Mobility Services for Climate Mitigation in Urban Areas: Case Studies of Baltic Countries and Germany', *Sustainability*, 13(8), p. 4127. Available at: https://doi.org/10.3390/su13084127.

Chang, T.-H. *et al.* (2014) 'Traffic Speed Estimation through Data Fusion from Heterogeneous Sources for First Response Deployment', *Journal of Computing in Civil Engineering* [Preprint]. Available at: https://doi.org/10.1061/(ASCE)CP.1943-5487.0000379.

Chaturvedi, S. *et al.* (2022) 'A comparative assessment of SARIMA, LSTM RNN and Fb Prophet models to forecast total and peak monthly energy demand for India', *Energy Policy*, 168, p. 113097. Available at: https://doi.org/10.1016/j.enpol.2022.113097.

Che, H. (2021) 'Multi-sensor Data Fusion Method Based on ARIMA-LightGBM for AGV Positioning', in *2021 5th International Conference on Robotics and Automation Sciences (ICRAS)*. *2021 5th International Conference on Robotics and Automation Sciences (ICRAS)*, pp. 272–276. Available at: https://doi.org/10.1109/ICRAS52289.2021.9476452.

Che Husin, M.S. and Abdul Nasir, A.S. (2021) 'Automatic Passengers Counting System Using Images Processing Based on YCbCr and HSV Colour Spaces Analysis', in Z. Md Zain et al. (eds) *Proceedings of the 11th National Technical Seminar on*

Unmanned System Technology 2019. Singapore: Springer Nature (Lecture Notes in Electrical Engineering), pp. 853–872. Available at: https://doi.org/10.1007/978-981-15-5281-6_62.

Chen, C. *et al.* (2021) 'An Edge Traffic Flow Detection Scheme Based on Deep Learning in an Intelligent Transportation System', *IEEE Transactions on Intelligent Transportation Systems*, 22(3), pp. 1840–1852. Available at: https://doi.org/10.1109/TITS.2020.3025687.

Chen, S.-Z. *et al.* (2018) 'Development of a Bridge Weigh-in-Motion System Based on Long-Gauge Fiber Bragg Grating Sensors', *Journal of Bridge Engineering*, 23. Available at: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29BE.1943-5592.0001283 (Accessed: 26 April 2023).

Chen, T. and Guestrin, C. (2016) 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery (KDD '16), pp. 785–794. Available at: https://doi.org/10.1145/2939672.2939785.

Chen, Y., Tang, Z. and Yu, J. (2012) 'Using multi-sensor data fusion to predict dangerous states of LNG transport tank', in *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*. *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 278–280. Available at: https://doi.org/10.1109/ICACI.2012.6463167.

Cheng, J. *et al.* (2024) 'Forecasting Bitcoin prices using artificial intelligence: Combination of ML, SARIMA, and Facebook Prophet models', *Technological Forecasting and Social Change*, 198, p. 122938. Available at: https://doi.org/10.1016/j.techfore.2023.122938.

Chiara Appendino (2019) 'Giallo e blu. I colori della Città di Torino', *Facebook*, 21 February. Available at: https://www.facebook.com/photo/?fbid=2355178827826053&set=a.1128335107177104 (Accessed: 14 March 2024).

Chicco, D., Warrens, M.J. and Jurman, G. (2021) 'The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation', *PeerJ Computer Science*, 7, p. e623. Available at: https://doi.org/10.7717/peerj-cs.623.

Città metropolitana di Torino (TO) (2024) *amministrazionicomunali.it*. Available at: https://www.amministrazionicomunali.it/citta-metropolitana-di-torino (Accessed: 13 March 2024).

Comune di Torino (TO) (2024) *amministrazionicomunali.it*. Available at: https://www.amministrazionicomunali.it/piemonte/torino (Accessed: 13 March 2024).

Coulaud, R., Keribin, C. and Stoltz, G. (2022) 'One-Station-Ahead Forecasting of Dwell Time, Arrival Delay and Passenger Flows on Trains Equipped with Automatic Passenger Counting (APC) Device', in.

Cyril, A., Mulangi, R.H. and George, V. (2018) 'Modelling and Forecasting Bus Passenger Demand using Time Series Method', in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 460–466. Available at: https://doi.org/10.1109/ICRITO.2018.8748443.

Dasarathy, B.V. (1997) 'Sensor fusion potential exploitation-innovative architectures and illustrative applications', *Proceedings of the IEEE*, 85(1), pp. 24–38. Available at: https://doi.org/10.1109/5.554206.

De Potter, P. *et al.* (2012) 'Passenger Counting in Public Rail Transport - Using Head-shoulder Contour Tracking', in *VISAPP 2012 - Proceedings of the International Conference on Computer Vision Theory and Applications*. *International Conference on Computer Vision Theory and Applications*, Ghent, Belgium, pp. 705–708.

Deisenroth, M.P., Faisal, A.A. and Ong, C.S. (2020) *Mathematics for Machine Learning*. Cambridge ; New York, NY: Cambridge University Press. Available at: https://mml-book.github.io/.

Dhou, S. *et al.* (2022) 'An IoT Machine Learning-Based Mobile Sensors Unit for Visually Impaired People', *Sensors*, 22(14), p. 5202. Available at: https://doi.org/10.3390/s22145202.

Diana Abraham, C. and Aravinth, J. (2018) 'Fusion Methods for Hyperspectral Image and LIDAR Data at Pixel-Level', *2018 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2018* [Preprint]. Available at: https://doi.org/10.1109/WISPNET.2018.8538460.

Dimitrakopoulos, G., Uden, L. and Varlamis, I. (2020) 'Intelligent transport systems and smart mobility', in G. Dimitrakopoulos, L. Uden, and I. Varlamis (eds) *The Future of Intelligent Transport Systems*. Elsevier, pp. 199–205. Available at: https://doi.org/10.1016/B978-0-12-818281-9.00018-8.

Drozdowski, P. *et al.* (2022) 'Signal-level fusion for indexing and retrieval of facial biometric data', *IET Biometrics*, 11(2), pp. 141–156. Available at: https://doi.org/10.1049/BME2.12063.

Eskandari, Z. and Ayughi, F. (2011) *Data Aggregation Tree Construction: Algorithms and Challenges*, *Wireless Sensor Networks*. IntechOpen. Available at: https://doi.org/10.5772/38730.

European Commission (2010) ‘Evaluation of Directive 2010/40/EU on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport, s.l.: s.n’.

European Commission (2016) ‘A European strategy on Cooperative Intelligent Transport Systems, a milestone towards cooperative, connected and automated mobility’. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52016DC0766 (Accessed: 21 April 2023).

European Commission (2021) ‘Horizon Europe: The EU research & innovation programme 2021-2027’, 19 March. Available at: https://research-and-innovation.ec.europa.eu/funding/funding-opportunities/funding-programmes-and-open-calls/horizon-europe_en (Accessed: 5 February 2024).

European Parliament (2018) ‘A European strategy on Cooperative Intelligent Transport Systems European Parliament resolution of 13 March 2018 on a European strategy on Cooperative Intelligent Transport Systems (2017/2067(INI))’. Available at: https://www.europarl.europa.eu/doceo/document/TA-8-2018-0063_EN.html (Accessed: 21 April 2023).

Fang, Z. *et al.* (2016) ‘Mining Popular Mobility Patterns from User GPS Trajectories’, in *2016 9th International Conference on Service Science (ICSS)*. *2016 9th International Conference on Service Science (ICSS)*, pp. 180–181. Available at: https://doi.org/10.1109/ICSS.2016.33.

Faouzi, N.E.E. and Klein, L.A. (2016) ‘Data Fusion for ITS: Techniques and Research Needs’, *Transportation Research Procedia*, 15, pp. 495–512. Available at: https://doi.org/10.1016/J.TRPRO.2016.06.042.

Farr, J.V. and Faber, I.J. (2023) *Engineering Economics of Life Cycle Cost Analysis*. 2nd edn. CRC Press. Available at: https://doi.org/10.1201/9781003254782.

Fernandes, B. *et al.* (2018) ‘Crowd Sensing for Urban Security in Smart Cities’, *JUCS - Journal of Universal Computer Science*, 24(3), pp. 302–321. Available at: https://doi.org/10.3217/jucs-024-03-0302.

Filippi, C. *et al.* (2023) ‘Trends in passenger transport optimisation’, *International Transactions in Operational Research*, 30(6), pp. 3057–3086. Available at: https://doi.org/10.1111/itor.13300.

Fiorini, M. and Lin, J.-C. (2015) ‘The importance of ITS in urban movement’, in *Clean Mobility and Intelligent Transport Systems. s.l.:Institution of Engineering and Technology*.

Fort, A., Peruzzi, G. and Pozzebon, A. (2021) ‘Quasi-Real Time Remote Video Surveillance Unit for LoRaWAN-based Image Transmission’, in *2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT)*. *2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT*

(MetroInd4.0&IoT), pp. 588–593. Available at: https://doi.org/10.1109/MetroInd4.0IoT51437.2021.9488519.

Gallina, A., Dalla Chiara, B. and Vallan, A. (2022) 'Theory and practice of Weigh in Motion (WIM) based on fibre Bragg gratings: lab tests for smart roads purposes', *European Transport/Trasporti Europei*, (88), pp. 1–11. Available at: https://doi.org/10.48295/ET.2022.88.5.

Gallo, F., Sacco, N. and Corman, F. (2023) 'Network-Wide Public Transport Occupancy Prediction Framework With Multiple Line Interactions', *IEEE Open Journal of Intelligent Transportation Systems*, 4, pp. 815–832. Available at: https://doi.org/10.1109/OJITS.2023.3331447.

GeeksforGeeks (2023) *Python - Coefficient of Determination-R2 score*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/python-coefficient-of-determination-r2-score/ (Accessed: 23 June 2024).

Geetha, S. *et al.* (2024) 'Time Series Financial Market Forecasting Based on Machine Learning', in *2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA)*. *2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA)*, pp. 1–5. Available at: https://doi.org/10.1109/AIMLA59606.2024.10531346.

Gerland, H.E. and Sutter, K. (2013) 'Automatic Passenger Counting (APC): Infra-Red Motion Analyzer for accurate counts in stations and rail, light-rail and bus operations'. INIT GmbH Innovations in Transportation. Available at: https://www.yumpu.com/en/document/read/11484755/automatic-passenger-counting-apc-infra-red-motion-init (Accessed: 25 April 2023).

Ghahramani, Z. (2012) 'Bayesian Modelling'. *MLSS 2012 - University of Cambridge*, UK. Available at: https://mlg.eng.cam.ac.uk/zoubin/talks/lect1bayes.pdf.

Gharge, S. *et al.* (2023) 'Real-time Object Detection using Haar Cascade Classifier for Robot Cars', in *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*. *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 64–70. Available at: https://doi.org/10.1109/ICESC57686.2023.10193401.

Ghosh, I. and Jana, R.K. (2024) 'Clean energy stock price forecasting and response to macroeconomic variables: A novel framework using Facebook's Prophet, NeuralProphet and explainable AI', *Technological Forecasting and Social Change*, 200, p. 123148. Available at: https://doi.org/10.1016/j.techfore.2023.123148.

Giuffrida, L., Masera, G. and Martina, M. (2023) 'A Survey of Automotive Radar and Lidar Signal Processing and Architectures', *Chips*, 2(4), pp. 243–261. Available at: https://doi.org/10.3390/chips2040015.

Gong, L. *et al.* (2022) 'Model Predictive Control-Based Real-Time Optimal Charging of Electric Vehicle Aggregators', in *2022 IEEE Power & Energy Society General Meeting (PESGM)*. *2022 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5. Available at: https://doi.org/10.1109/PESGM48719.2022.9916951.

Grgurević, I., Juršić, K. and Rajič, V. (2022) 'Overview of Wi-Fi-Based Automatic Passenger Counting Solutions in Public Urban Transport', in L. Knapcikova et al. (eds) *Sustainable Management of Manufacturing Systems in Industry 4.0*. Cham: Springer International Publishing (EAI/Springer Innovations in Communication and Computing). Available at: https://doi.org/10.1007/978-3-030-90462-3_12.

Grönman, J. *et al.* (2019) 'People counting in a public event—use case: Free-to-ride bus', *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings*, pp. 1055–1059. Available at: https://doi.org/10.23919/MIPRO.2019.8756921.

GTT - Gruppo Torinese Trasporti (2022) *Carta della Mobilità GTT 2022*. Turin, Italy. Available at: https://www.gtt.to.it/cms/avvisi-e-informazioni-di-servizio/varie/9727-carta-della-mobilita-gtt-e-online-l-edizione-2022 (Accessed: 20 August 2022).

Guo, Y. and Zhang, W. (2022) 'Real-time monitoring model of passenger flow congestion of subway station based on improved YOLOv3', in *International Conference on Smart Transportation and City Engineering (STCE 2022)*. *International Conference on Smart Transportation and City Engineering (STCE 2022)*, SPIE, pp. 773–783. Available at: https://doi.org/10.1117/12.2659143.

Hadas, Y. *et al.* (2022) 'Assessing public transport passenger attitudes towards a dynamic fare model based on in-vehicle crowdedness levels and additional waiting time', *International Journal of Transportation Science and Technology* [Preprint]. Available at: https://doi.org/10.1016/j.ijtst.2022.08.003.

Han, J. (2012) *Multi-Sensor Data Fusion for Travel Time Estimation*. London: Imperial College London.

Hangli, G. *et al.* (2022) 'Multi-weighted Graphs Learning for Passenger Count Prediction on Railway Network', in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*. *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 374–382. Available at: https://doi.org/10.1109/COMPSAC54236.2022.00059.

Hauke, J. and Kossowski, T. (2011) 'Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data', *Quaestiones Geographicae*, 30(2), pp. 87–93. Available at: https://doi.org/10.2478/v10117-011-0021-1.

He, Q. *et al.* (2017) 'A parallel clustering and test partitioning techniques based mining trajectory algorithm for moving objects', in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*.

2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 455–462. Available at: https://doi.org/10.1109/FSKD.2017.8393312.

HELLA Aglaia (2022) *Advanced People Sensor APS-B*. Available at: https://people-sensing.com/public-transport/#products (Accessed: 1 July 2022).

Hensher, D.A. (2020) 'Chapter 23 - A review of objective and subjective measures of crowding in public transport', in D.A. Hensher (ed.) *Bus Transport*. Elsevier, pp. 281–294. Available at: https://doi.org/10.1016/B978-0-12-820132-9.00023-6.

Hepner, E. *et al.* (2021) 'Optimization of the traffic flow in public transport through C-ITS', in *2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. *2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 1–5. Available at: https://doi.org/10.1109/MT-ITS49943.2021.9529300.

Hoang, M.X., Zheng, Y. and Singh, A.K. (2016) 'FCCF: forecasting citywide crowd flows based on big data', in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, USA: Association for Computing Machinery (SIGSPACIAL '16), pp. 1–10. Available at: https://doi.org/10.1145/2996913.2996934.

Hongmei, L. *et al.* (2020) 'Object Tracking in Video Sequence based on Kalman filter', in *2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC)*. *2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC)*, pp. 106–110. Available at: https://doi.org/10.1109/ICCEIC51584.2020.00029.

Hoppe, J. *et al.* (2023) 'Improving the Prediction of Passenger Numbers in Public Transit Networks by Combining Short-Term Forecasts With Real-Time Occupancy Data', *IEEE Open Journal of Intelligent Transportation Systems*, 4, pp. 153–174. Available at: https://doi.org/10.1109/OJITS.2023.3251564.

Hou, X., Wang, Y. and Chau, L.P. (2019) 'Vehicle tracking using deep SORT with low confidence track filtering', *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2019* [Preprint]. Available at: https://doi.org/10.1109/AVSS.2019.8909903.

Hsu, Y.-W., Chen, Y.-W. and Perng, J.-W. (2020a) 'Estimation of the Number of Passengers in a Bus Using Deep Learning', *Sensors*, 20(8), p. 2178. Available at: https://doi.org/10.3390/s20082178.

Hsu, Y.W., Wang, T.Y. and Perng, J.W. (2020b) 'Passenger flow counting in buses based on deep learning using surveillance video', *Optik*, 202, p. 163675. Available at: https://doi.org/10.1016/J.IJLEO.2019.163675.

Hu, R., Chiu, Y.-C. and Hsieh, C.-W. (2020) 'Crowding prediction on mass rapid transit systems using a weighted bidirectional recurrent neural network', *IET Intelligent*

Transport Systems, 14(3), pp. 196–203. Available at: https://doi.org/10.1049/iet-its.2018.5542.

Iliopoulou, C. and Kepaptsoglou, K. (2019) 'Combining ITS and optimization in public transportation planning: state of the art and future research paths', *European Transport Research Review 2019 11:1*, 11(1), pp. 1–16. Available at: https://doi.org/10.1186/S12544-019-0365-5.

IRIS intelligent sensing (2016) *Rhein-Neckar-Verkehr-GmbH (rnv)*. Available at: https://www.iris-sensing.com/references/details/rhein-neckar-verkehr-gmbh-rnv/ (Accessed: 13 April 2022).

Islam, Md.T., Ahmad, M. and Bappy, A.S. (2020) 'Real-Time Family Member Recognition Using Raspberry Pi for Visually Impaired People', in *2020 IEEE Region 10 Symposium (TENSYMP)*. *2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 78–81. Available at: https://doi.org/10.1109/TENSYMP50017.2020.9230937.

Iyappan, P. *et al.* (2024) 'An Optimized Combinatorial Approach For Stock Market Prediction Considering User Preferences', in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS)*. *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS)*, pp. 492–497. Available at: https://doi.org/10.1109/ICC-ROBINS60238.2024.10533891.

Jabbar, W.A. *et al.* (2021) 'An IoT Raspberry Pi-based parking management system for smart campus', *Internet of Things*, 14, p. 100387. Available at: https://doi.org/10.1016/j.iot.2021.100387.

Jain, S. *et al.* (2021) 'Sustainable Infrastructure Theories and Models', in *Digital Cities Roadmap*. John Wiley & Sons, Ltd, pp. 97–126. Available at: https://doi.org/10.1002/9781119792079.ch3.

James, G. *et al.* (2021) 'Linear Model Selection and Regularization', in G. James et al. (eds) *An Introduction to Statistical Learning: with Applications in R*. 2nd edn. New York, NY: Springer US (Springer Texts in Statistics), pp. 225–288. Available at: https://doi.org/10.1007/978-1-0716-1418-1_6.

Jenelius, E. (2020a) 'Data-Driven Metro Train Crowding Prediction Based on Real-Time Load Data', *IEEE Transactions on Intelligent Transportation Systems*, 21(6), pp. 2254–2265. Available at: https://doi.org/10.1109/TITS.2019.2914729.

Jenelius, E. (2020b) 'Personalized predictive public transport crowding information with automated data sources', *Transportation Research Part C: Emerging Technologies*, 117, p. 102647. Available at: https://doi.org/10.1016/j.trc.2020.102647.

Kabanov, A.S., Azarov, V.N. and Mayboroda, V.P. (2019) 'An Analysis of the Use and Difficulties in Introducing Information Technology and Information Systems in

Transport and the Transport Infrastructure', in *2019 International Conference 'Quality Management, Transport and Information Security, Information Technologies' (IT&QM&IS)*. *2019 International Conference 'Quality Management, Transport and Information Security, Information Technologies' (IT&QM&IS)*, pp. 192–196. Available at: https://doi.org/10.1109/ITQMIS.2019.8928298.

Kamath, V. and A., R. (2021) 'Performance Analysis of the Pretrained EfficientDet for Real-time Object Detection on Raspberry Pi', in *2021 International Conference on Circuits, Controls and Communications (CCUBE)*. *2021 International Conference on Circuits, Controls and Communications (CCUBE)*, pp. 1–6. Available at: https://doi.org/10.1109/CCUBE53681.2021.9702741.

Kavitha, N. and Chandrappa, D.N. (2017) 'Comparative Review on Video Based Vehicular Traffic Data Collection for Intelligent Transport System', in *2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*. *2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, pp. 260–264. Available at: https://doi.org/10.1109/ICRAECT.2017.31.

Kavya D (2023) 'Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning', *Medium*, 16 February. Available at: https://medium.com/@Kavya2099/optimizing-performance-selectkbest-for-efficient-feature-selection-in-machine-learning-3b635905ed48 (Accessed: 2 May 2024).

Ke, G. *et al.* (2017a) 'LightGBM: A Highly Efficient Gradient Boosting Decision Tree', in *Advances in Neural Information Processing Systems*. Curran Associates, Inc. Available at: https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html (Accessed: 6 June 2024).

Ke, G. *et al.* (2017b) 'LightGBM: A Highly Efficient Gradient Boosting Decision Tree', in *Advances in Neural Information Processing Systems*. Curran Associates, Inc. Available at: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html (Accessed: 16 March 2024).

Ke, X. *et al.* (2019) 'Multi-Dimensional Traffic Congestion Detection Based on Fusion of Visual Features and Convolutional Neural Network', *IEEE Transactions on Intelligent Transportation Systems*, 20(6), pp. 2157–2170. Available at: https://doi.org/10.1109/TITS.2018.2864612.

Khaire, U.M. and Dhanalakshmi, R. (2022) 'Stability of feature selection algorithm: A review', *Journal of King Saud University - Computer and Information Sciences*, 34(4), pp. 1060–1073. Available at: https://doi.org/10.1016/j.jksuci.2019.06.012.

Kieu, L.M. and Cai, C. (2018) 'Stochastic collective model of public transport passenger arrival process', *IET Intelligent Transport Systems*, 12(9), pp. 1027–1035. Available at: https://doi.org/10.1049/iet-its.2018.0085.

Klein, L. (2022) 'Chapter 2: Sensor and Data Fusion in Traffic Management', in, pp. 11–98.

Klein, L.A. (2019) '1. Introduction', in *Sensor and Data Fusion for Intelligent Transportation Systems*. SPIE. Available at: https://app.knovel.com/hotlink/khtml/id:kt012EH9W1/sensor-data-fusion-intelligent/introduction.

Kniess, J., Rutke, J.C. and Castañeda, W.A.C. (2021) 'An IoT Transport Architecture for Passenger Counting: A Real Implementation', in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Bordeaux, France, pp. 613–617.

Korstanje, J. (2021) *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR*. Berkeley, CA: Apress. Available at: https://doi.org/10.1007/978-1-4842-7150-6.

Kuchár, P. *et al.* (2021) 'Noninvasive Passenger Detection Comparison Using Thermal Imager and IP Cameras', *Sustainability*, 13(22), p. 12928. Available at: https://doi.org/10.3390/su132212928.

Kuchár, P. *et al.* (2023) 'Passenger Occupancy Estimation in Vehicles: A Review of Current Methods and Research Challenges', *Sustainability*, 15(2), p. 1332. Available at: https://doi.org/10.3390/su15021332.

Kumar Singh, A., Singh, D. and Goyal, M. (2021) 'People Counting System Using Python', *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, pp. 1750–1754. Available at: https://doi.org/10.1109/ICCMC51019.2021.9418290.

Lee, R.J., Sener, I.N. and Mullins, J.A. (2016) 'An evaluation of emerging data collection technologies for travel demand modeling: from research to practice', *Transportation Letters*, 8(4), pp. 181–193. Available at: https://doi.org/10.1080/19427867.2015.1106787.

Lenatti, M. *et al.* (2023) 'Dual-View Single-Shot Multibox Detector at Urban Intersections: Settings and Performance Evaluation', *Sensors*, 23(6), p. 3195. Available at: https://doi.org/10.3390/s23063195.

Li, J. *et al.* (2021) 'Obstacle detection and tracking algorithm based on multi-lidar fusion in urban environment', *IET Intelligent Transport Systems*, 15(11), pp. 1372–1387. Available at: https://doi.org/10.1049/ITR2.12105.

Li, Q. *et al.* (2014) 'A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios', *IEEE Transactions on Vehicular Technology*, 63(2), pp. 540–555. Available at: https://doi.org/10.1109/TVT.2013.2281199.

Li, X. and Xie, M. (2021) 'Short-term passenger volume forecast and model analysis of Beijing public transport', in *Fifth International Conference on Traffic Engineering and Transportation System (ICTETS 2021)*. *Fifth International Conference on Traffic Engineering and Transportation System (ICTETS 2021)*, SPIE, pp. 1423–1429. Available at: https://doi.org/10.1117/12.2619923.

Liang, Y., Qian, X. and Zhu, L. (2021) 'Towards Better Railway Service: Passengers Counting in Railway Compartment', *IEEE Transactions on Circuits and Systems for Video Technology*, 31(2), pp. 439–451. Available at: https://doi.org/10.1109/TCSVT.2020.2979984.

Liciotti, D. *et al.* (2017) 'An Intelligent RGB-D Video System for Bus Passenger Counting', in W. Chen et al. (eds) *Intelligent Autonomous Systems 14*. Cham: Springer International Publishing (Advances in Intelligent Systems and Computing), pp. 473–484. Available at: https://doi.org/10.1007/978-3-319-48036-7_34.

Lin, K. *et al.* (2019) 'Different Levels Multi-source Remote Sensing Image Fusion', *ICSIDP 2019 - IEEE International Conference on Signal, Information and Data Processing 2019* [Preprint]. Available at: https://doi.org/10.1109/ICSIDP47821.2019.9173281.

Lin, P. *et al.* (2020) 'Analysing the relationship between weather, built environment, and public transport ridership', *IET Intelligent Transport Systems*, 14(14), pp. 1946–1954. Available at: https://doi.org/10.1049/iet-its.2020.0469.

Linford, S. *et al.* (2019) 'Data Analysis on Big Data Applications with Small Samples and Incomplete Information', in *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 146–151. Available at: https://doi.org/10.1109/CSCWD.2019.8791927.

Liu, G. *et al.* (2017) 'Passenger flow estimation based on convolutional neural network in public transportation system', *Knowledge-Based Systems*, 123, pp. 102–115. Available at: https://doi.org/10.1016/J.KNOSYS.2017.02.016.

Liu, J. *et al.* (2020) 'Urban big data fusion based on deep learning: An overview', *Information Fusion*, 53, pp. 123–133. Available at: https://doi.org/10.1016/j.inffus.2019.06.016.

Liu, L. *et al.* (2020) 'A Technology for Automatically Counting Bus Passenger Based on YOLOv2 and MIL Algorithm', in *2020 IEEE 5th International Conference on*

Image, Vision and Computing (ICIVC), pp. 166–170. Available at: https://doi.org/10.1109/ICIVC50857.2020.9177434.

Liu, L. and Chen, R.-C. (2017) 'A MRT Daily Passenger Flow Prediction Model with Different Combinations of Influential Factors', in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 601–605. Available at: https://doi.org/10.1109/WAINA.2017.19.

Llinas, J. *et al.* (2004) *Revisiting the JDL data fusion model II*, *Proceedings of the Seventh International Conference on Information Fusion, FUSION 2004*. SPACE AND NAVAL WARFARE SYSTEMS COMMAND SAN DIEGO CA.

Lu, F. *et al.* (2021) 'Ultra-Short-Term Prediction of EV Aggregator's Demond Response Flexibility Using ARIMA, Gaussian-ARIMA, LSTM and Gaussian-LSTM', in *2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST)*. *2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, pp. 1775–1781. Available at: https://doi.org/10.1109/IAECST54258.2021.9695933.

Lu, M. *et al.* (2021) 'Research on Traffic Information Extraction Approach Based on Image Deep Learning', in *2021 6th International Conference on Transportation Information and Safety (ICTIS)*. *2021 6th International Conference on Transportation Information and Safety (ICTIS)*, pp. 1275–1279. Available at: https://doi.org/10.1109/ICTIS54573.2021.9798479.

Lu, X.-L., Hsu, H.-J. and Chu, W.C.-C. (2023) 'Public Bicycle Flow Forecasting using Spatial and Temporal Graph Neural Network', in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 490–499. Available at: https://doi.org/10.1109/COMPSAC57700.2023.00071.

Luo, R.C., Yih, C.C. and Su, K.L. (2002) 'Multisensor fusion and integration: Approaches, applications, and future research directions', *IEEE Sensors Journal*, 2(2), pp. 107–119. Available at: https://doi.org/10.1109/JSEN.2002.1000251.

M, S.A., S, R.R. and Ranjana, P. (2024) 'Walmart Sales Forecasting Using Time Series Analysis', in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pp. 1–5. Available at: https://doi.org/10.1109/ADICS58448.2024.10533547.

Macheret, P.D., Savchuk, R.R. and Shkuratov, G.I. (2021) 'Intelligent Transport Systems: Analysis of the Current State and Prospects of Development', in *2021 International Conference on Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*. *2021 International Conference on Quality Management, Transport and Information Security, Information*

Technologies (IT&QM&IS), pp. 234–237. Available at: https://doi.org/10.1109/ITQMIS53292.2021.9642830.

Mahapatra, S. (2018) *Why Deep Learning over Traditional Machine Learning?*, *Medium*. Available at: https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063 (Accessed: 27 September 2022).

Mangiaracina, R. *et al.* (2017) 'A comprehensive view of intelligent transport systems for urban smart mobility', *International Journal of Logistics Research and Applications*, 20(1), pp. 39–52. Available at: https://doi.org/10.1080/13675567.2016.1241220.

Mccarthy, C. *et al.* (2021) 'A Field Study of Internet of Things-Based Solutions for Automatic Passenger Counting', *IEEE Open Journal of Intelligent Transportation Systems*, 2, pp. 384–401. Available at: https://doi.org/10.1109/OJITS.2021.3111052.

Mehmood, U. *et al.* (2019) 'Occupancy Estimation using WiFi: A Case Study for Counting Passengers on Busses', in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp. 165–170. Available at: https://doi.org/10.1109/WF-IoT.2019.8767350.

Mei, Z. *et al.* (2023) 'Attention mechanism-based model for short-term bus traffic passenger volume prediction', *IET Intelligent Transport Systems*, 17(4), pp. 767–779. Available at: https://doi.org/10.1049/itr2.12302.

Meng, T. *et al.* (2020) 'A survey on machine learning for data fusion', *Information Fusion*, 57(2), pp. 115–129. Available at: https://doi.org/10.1016/j.inffus.2019.12.001.

Mind Tools (2024) *Decision Matrix Analysis - Making a Decision by Weighing Up Different Factors*, *MindTools*. Available at: https://www.mindtools.com/aksic2i/decision-matrix-analysis (Accessed: 22 June 2024).

Moreno Rendon, W.D. *et al.* (2023) 'Passenger Counting in Mass Public Transport Systems using Computer Vision and Deep Learning', *IEEE Latin America Transactions*, 21(4), pp. 537–545. Available at: https://doi.org/10.1109/TLA.2023.10128885.

Munim, Z.H. *et al.* (2023) 'Forecasting container throughput of major Asian ports using the Prophet and hybrid time series models', *The Asian Journal of Shipping and Logistics*, 39(2), pp. 67–77. Available at: https://doi.org/10.1016/j.ajsl.2023.02.004.

Munteanu, D. *et al.* (2022) 'Sea Mine Detection Framework Using YOLO, SSD and EfficientDet Deep Learning Models', *Sensors*, 22(23), p. 9536. Available at: https://doi.org/10.3390/s22239536.

Myrvoll, T.A. *et al.* (2017) 'Counting public transport passenger using WiFi signatures of mobile devices', in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6. Available at: https://doi.org/10.1109/ITSC.2017.8317687.

Nakashima, H., Arai, I. and Fujikawa, K. (2019) 'Passenger Counter Based on Random Forest Regressor Using Drive Recorder and Sensors in Buses', in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 561–566. Available at: https://doi.org/10.1109/PERCOMW.2019.8730761.

Nasir, A.S.A., Gharib, N.K.A. and Jaafar, H. (2018) 'Automatic Passenger Counting System Using Image Processing Based on Skin Colour Detection Approach', *2018 International Conference on Computational Approach in Smart Systems Design and Applications, ICASSDA 2018* [Preprint]. Available at: https://doi.org/10.1109/ICASSDA.2018.8477628.

Nguyen, H. *et al.* (2018) 'Deep learning methods in transportation domain: a review', *IET Intelligent Transport Systems*, 12(9), pp. 998–1004. Available at: https://doi.org/10.1049/iet-its.2018.0064.

Nitti, M. *et al.* (2020) 'iABACUS: A Wi-Fi-Based Automatic Bus Passenger Counting System', *Energies*, 13(6), p. 1446. Available at: https://doi.org/10.3390/en13061446.

Nogueira, V. *et al.* (2019) 'RetailNet: A deep learning approach for people counting and hot spots detection in retail stores', *Proceedings - 32nd Conference on Graphics, Patterns and Images, SIBGRAPI 2019*, pp. 155–162. Available at: https://doi.org/10.1109/SIBGRAPI.2019.00029.

Nowruzi, F.E. *et al.* (2019) 'In-Vehicle Occupancy Detection With Convolutional Networks on Thermal Images', in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 941–948. Available at: https://doi.org/10.1109/CVPRW.2019.00124.

Olivo, A., Maternini, G. and Barabino, B. (2020) 'Empirical Study on the Accuracy and Precision of Automatic Passenger Counting in European Bus Services', *The Open Transportation Journal*, 13(1), pp. 250–260. Available at: https://doi.org/10.2174/1874447801913010250.

Oroceo, P.P. *et al.* (2022) 'Optimizing Face Recognition Inference with a Collaborative Edge–Cloud Network', *Sensors*, 22(21), p. 8371. Available at: https://doi.org/10.3390/s22218371.

Pan, Y.A. *et al.* (2024) 'A fundamental diagram based hybrid framework for traffic flow estimation and prediction by combining a Markovian model with deep learning',

Expert Systems with Applications, 238, p. 122219. Available at: https://doi.org/10.1016/j.eswa.2023.122219.

Paneru, S. and Jeelani, I. (2021) 'Computer vision applications in construction: Current state, opportunities & challenges', *Automation in Construction*, 132, p. 103940. Available at: https://doi.org/10.1016/J.AUTCON.2021.103940.

Panetta, K. *et al.* (2021) 'Artificial Intelligence for Text-Based Vehicle Search, Recognition, and Continuous Localization in Traffic Videos', *AI*, 2(4), pp. 684–704. Available at: https://doi.org/10.3390/ai2040041.

Paradkar, S. (2016) *Cracking the IT Architect Interview*. United Kingdom: Packt Publishing Ltd.

Pariota, L. *et al.* (2019) 'Green Light Optimal Speed Advisory: a C-ITS to improve mobility and pollution', in *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, pp. 1–6. Available at: https://doi.org/10.1109/EEEIC.2019.8783573.

Patel, M. *et al.* (2022) 'Visual Analysis of Thane Intelligent Transport System', in *2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC)*. *2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC)*, pp. 58–64. Available at: https://doi.org/10.1109/IIHC55949.2022.10059995.

Pavani, K. and Sriramya, P. (2022) 'Comparison of KNN, ANN, CNN and YOLO algorithms for detecting the accurate traffic flow and build an Intelligent Transportation System', in *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*. *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, pp. 628–633. Available at: https://doi.org/10.1109/ICIPTM54933.2022.9753900.

Pavlidis, I. *et al.* (1999) 'A near-infrared fusion scheme for automatic detection of vehicle passengers', in *Proceedings IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications (CVBVS'99)*. *Proceedings IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications (CVBVS'99)*, pp. 41–48. Available at: https://doi.org/10.1109/CVBVS.1999.781093.

Pavlyuk, D. (2017) 'Short-term Traffic Forecasting Using Multivariate Autoregressive Models', *Procedia Engineering*, 178, pp. 57–66. Available at: https://doi.org/10.1016/j.proeng.2017.01.062.

Pei, Y., Liu, H. and Bei, Q. (2021) 'Collision-Line Counting Method Using DeepSORT to Count Pedestrian Flow Density and Hungary Algorithm', in *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology*

(ICCASIT). *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, pp. 621–626. Available at: https://doi.org/10.1109/ICCASIT53235.2021.9633356.

Perallos, A. *et al.* (2015) *Intelligent Transport Systems: Technologies and Applications*. Wiley.

Pinna, I. and Dalla Chiara, B. (2010) 'Automatic passenger counting and vehicle load monitoring', *Ingegneria Ferroviaria*, 65, pp. 101–138.

Pouyanfar, Samira *et al.* (2018) 'A Survey on Deep Learning', *ACM Computing Surveys (CSUR)*, 51(5), p. 92. Available at: https://doi.org/10.1145/3234150.

Pronello, C., Baratti, L. and Anbarasan, D. (2024) 'Benchmarking the Functional, Technical, and Business Characteristics of Automated Passenger Counting Products', *Smart Cities*, 7(1), pp. 302–324. Available at: https://doi.org/10.3390/smartcities7010013.

Pronello, C. and Garzón Ruiz, X.R. (2023) 'Evaluating the Performance of Video-Based Automated Passenger Counting Systems in Real-World Conditions: A Comparative Study', *Sensors*, 23(18), p. 7719. Available at: https://doi.org/10.3390/s23187719.

Qiu, H. *et al.* (2021) 'Single Variable-Constrained NDT Matching in Traffic Data Collection Using a Laser-Based Detector', *IEEE Access*, 9, pp. 53123–53132. Available at: https://doi.org/10.1109/ACCESS.2021.3069905.

Rahaman, M.A. *et al.* (2022) 'Performance Analysis between YOLOv5s and YOLOv5m Model to Detect and Count Blood Cells: Deep Learning Approach', *ACM International Conference Proceeding Series*, pp. 316–322. Available at: https://doi.org/10.1145/3542954.3543000.

Rajput, P. and Chaturvedi, M. (2019) 'Automatic Detection of Bus-stops and Bus-crowdedness using Crowdsourced Data', in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 740–745. Available at: https://doi.org/10.1109/ITSC.2019.8917008.

Rajput, P., Chaturvedi, M. and Patel, V. (2020) 'Opportunistic sensing based detection of crowdedness in public transport buses', *Pervasive and Mobile Computing*, 68, p. 101246. Available at: https://doi.org/10.1016/j.pmcj.2020.101246.

Rajput, P., Chaturvedi, M. and Patel, V. (2022) 'Development of advanced public transportation system with limited infrastructure support', *Software Impacts*, 14, p. 100434. Available at: https://doi.org/10.1016/j.simpa.2022.100434.

Rakebrandt, A. (2013) 'Automatic Passenger Counting with expandable features - INIT'. *APC Webinar*. Available at: https://www.itscanada.ca/files/MOBILE-APC_ITSCanada_webinar.pdf (Accessed: 26 April 2023).

Ramgopal, I.A. and Manivannan, P.V. (2016) 'Development of multi-sensor data fusion technique for the automated Bus Rapid Transport System', in *2016 International Conference on Robotics: Current Trends and Future Challenges (RCTFC)*. *2016 International Conference on Robotics: Current Trends and Future Challenges (RCTFC)*, pp. 1–6. Available at: https://doi.org/10.1109/RCTFC.2016.7893409.

Ramzai, J. (2019) *Simple guide for ensemble learning methods*, *Medium*. Available at: https://towardsdatascience.com/simple-guide-for-ensemble-learning-methods-d87cc68705a2 (Accessed: 16 March 2024).

Recalde, G. *et al.* (2021) 'Low-cost in-bus people counting system for the reordering of urban passenger traffic based on artificial vision and neural networks', *2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pp. 1–4. Available at: https://doi.org/10.1109/CHILECON54041.2021.9703018.

Redmon, J. and Farhadi, A. (2018) 'YOLOv3: An Incremental Improvement', *arXiv* [Preprint]. Available at: https://doi.org/10.48550/ARXIV.1804.02767.

Ren, Y. *et al.* (2020) 'A hybrid integrated deep learning model for the prediction of citywide spatio-temporal flow volumes', *International Journal of Geographical Information Science*, 34(4), pp. 802–823. Available at: https://doi.org/10.1080/13658816.2019.1652303.

Rodrigue, J.-P. (2020) 'Chapter 8 : Urban transportation', in *The Geography of Transport Systems*. Routledge, pp. 283–319. Available at: https://transportgeography.org/.

Roncoli, C., Chandakas, E. and Kaparias, I. (2023) 'Estimating on-board passenger comfort in public transport vehicles using incomplete automatic passenger counting data', *Transportation Research Part C: Emerging Technologies*, 146, p. 103963. Available at: https://doi.org/10.1016/j.trc.2022.103963.

Ruiz-Aguilar, J.J., Turias, I.J. and Jiménez-Come, M.J. (2014) 'Hybrid approaches based on SARIMA and artificial neural networks for inspection time series forecasting', *Transportation Research Part E: Logistics and Transportation Review*, 67, pp. 1–13. Available at: https://doi.org/10.1016/j.tre.2014.03.009.

Safi, H. *et al.* (2017) 'An empirical comparison of four technology-mediated travel survey methods', *Journal of Traffic and Transportation Engineering (English Edition)*, 4(1), pp. 80–87. Available at: https://doi.org/10.1016/j.jtte.2015.12.003.

Sari Aslam, N. *et al.* (2021) 'ActivityNET: Neural networks to predict public transport trip purposes from individual smart card data and POIs', *Geo-spatial Information Science*, 24(4), pp. 711–721. Available at: https://doi.org/10.1080/10095020.2021.1985943.

Schmitt, M. and Zhu, X. (2016) 'Data Fusion and Remote Sensing – An Ever-Growing Relationship', *IEEE Geoscience and Remote Sensing Magazine*, 4, pp. 6–23. Available at: https://doi.org/10.1109/MGRS.2016.2561021.

Seidel, R. *et al.* (2022) 'NAPC: A Neural Algorithm for Automated Passenger Counting in Public Transport on a Privacy-Friendly Dataset', *IEEE Open Journal of Intelligent Transportation Systems*, 3, pp. 33–44. Available at: https://doi.org/10.1109/OJITS.2021.3139393.

Shakeel, A., Chong, D. and Wang, J. (2023) 'Load forecasting of district heating system based on improved FB-Prophet model', *Energy*, 278, p. 127637. Available at: https://doi.org/10.1016/j.energy.2023.127637.

Sharara, M., Ibrahim, M. and Chalhoub, G. (2019) 'Impact of Network Performance on GLOSA', in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6. Available at: https://doi.org/10.1109/CCNC.2019.8651787.

Shi, Y. *et al.* (2023) 'Research on fine classification of urban rail transit passengers based on two-step clustering-FCM fusion algorithm', in *International Conference on Algorithms, High Performance Computing, and Artificial Intelligence (AHPCAI 2023)*. *International Conference on Algorithms, High Performance Computing, and Artificial Intelligence (AHPCAI 2023)*, SPIE, pp. 47–52. Available at: https://doi.org/10.1117/12.3011526.

Shiao, Y.C. *et al.* (2017) 'Predicting passenger flow using different influence factors for Taipei MRT system', in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*. *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*, pp. 447–451. Available at: https://doi.org/10.1109/ICAwST.2017.8256497.

Sobrino, D.S. *et al.* (2023) 'Model Generation and Origin - Destination(OD) Matrix Forecasting for metro systems', in *2023 7th International Young Engineers Forum (YEF-ECE)*. *2023 7th International Young Engineers Forum (YEF-ECE)*, pp. 112–117. Available at: https://doi.org/10.1109/YEF-ECE58420.2023.10209325.

Song, J. *et al.* (2008) 'Infrared passenger flow collection system based on RBF neural net', in *2008 International Conference on Machine Learning and Cybernetics*. *2008 International Conference on Machine Learning and Cybernetics*, pp. 1277–1281. Available at: https://doi.org/10.1109/ICMLC.2008.4620601.

Soviany, P. and Ionescu, R.T. (2018) 'Optimizing the Trade-off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction', in *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, Timisoara, Romania, pp. 209–214. Available at: https://doi.org/10.1109/SYNASC.2018.00041.

Sun, R. *et al.* (2020) 'Robust IμGPS/VO Integration for Vehicle Navigation in GNSS Degraded Urban Areas', *IEEE Sensors Journal*, 20(17), pp. 10110–10122. Available at: https://doi.org/10.1109/JSEN.2020.2989332.

Swamynathan, M. (2019) 'Step 6: Deep and Reinforcement Learning', in M. Swamynathan (ed.) *Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python*. Berkeley, CA: Apress, pp. 383–442. Available at: https://doi.org/10.1007/978-1-4842-4947-5_6.

Tapeswitch (2020) *Transportation products*, *Treadle Mats, Passenger Signaling, Door Edges*. Available at: https://pdf.directindustry.com/pdf/tapeswitch/treadle-mats-passenger-signaling-door-edges/11682-72508.html (Accessed: 30 April 2023).

Tarantino, A. (2022) *Smart Manufacturing: The Lean Six Sigma Way*. John Wiley & Sons.

Thikshaja, U.K. and Paul, A. (2019) 'A Brief Review on Deep Learning and Types of Implementation for Deep Learning', in *Deep Learning and Neural Networks*. IGI Global, pp. 30–40. Available at: https://doi.org/10.4018/978-1-7998-0414-7.ch002.

Toto, E. *et al.* (2016) 'PULSE: A Real Time System for Crowd Flow Prediction at Metropolitan Subway Stations', in B. Berendt et al. (eds) *Machine Learning and Knowledge Discovery in Databases*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 112–128. Available at: https://doi.org/10.1007/978-3-319-46131-1_19.

Transport System Catapult (2017) *Modernising the National Travel Survey. WP1 - Review of Current and Future. Technologies and Data Sources Report*. UK.

Turnip, M. *et al.* (2020) 'Detection of Vehicle Maximum Speed with an Infrared Sensor Based on Raspberry Pi 3 b +', in *2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT)*. *2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT)*, pp. 317–320. Available at: https://doi.org/10.1109/MECnIT48290.2020.9166678.

USA department of transportation (2021) *ITS Transit Fact Sheet - Automatic Passenger Counter - ITS Professional Capacity Building Program*. Available at: https://www.pcb.its.dot.gov/factSheets/apc/apc_overview.aspx#page=tech (Accessed: 27 June 2022).

USA Federal Highway Administration (2014) *Office of Highway Policy Information - Policy | Federal Highway Administration*, *Office of Highway Policy Information*. Available at: https://www.fhwa.dot.gov/policyinformation/pubs/vdstits2007/04pt2.cfm (Accessed: 26 April 2023).

Vagropoulos, S.I. *et al.* (2016) 'Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting', in *2016 IEEE International Energy Conference (ENERGYCON)*. *2016 IEEE*

International Energy Conference (ENERGYCON), pp. 1–6. Available at: https://doi.org/10.1109/ENERGYCON.2016.7514029.

Valencia, I.J.C. *et al.* (2022) 'Overhead View Bus Passenger Detection and Counter using DeepSORT and Tiny-Yolo V4', in *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*. *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pp. 1–6. Available at: https://doi.org/10.1109/HNICEM57413.2022.10109539.

Varghese, D. (2018) 'Comparative Study on Classic Machine learning Algorithms , Part-2'. Available at: https://medium.com/@dannymvarghese/comparative-study-on-classic-machine-learning-algorithms-part-2-5ab58b683ec0.

VDV Industrieforum (2018) *Automatic Passenger Counting Systems (APCS): Recommendations for the Application of APCSs within Public Transport and Regional Rail Transport, version 2.1*, *VDV Recommendation 457*. VDV Recommendation 457.

Venkatesh, B. and Anuradha, J. (2019) 'A Review of Feature Selection and Its Methods', *Cybernetics and Information Technologies*, 19(1), pp. 3–26. Available at: https://doi.org/10.2478/cait-2019-0001.

Verma, J.P. and Verma, P. (2020) *Determining Sample Size and Power in Research Studies: A Manual for Researchers*. Singapore: Springer. Available at: https://doi.org/10.1007/978-981-15-5204-5.

Vieira, D.F. and Júnior, H.G. (2022) 'Public Transportation Passengers Accounting at University by IoT Device', in *2022 Symposium on Internet of Things (SIoT)*. *2022 Symposium on Internet of Things (SIoT)*, pp. 1–5. Available at: https://doi.org/10.1109/SIoT56383.2022.10070138.

Vignarca, D. *et al.* (2021) 'Improved Person Counting Performance Using Kalman Filter Based on Image Detection and Tracking', in *2021 AEIT International Conference on Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*. *2021 AEIT International Conference on Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, pp. 1–6. Available at: https://doi.org/10.23919/AEITAUTOMOTIVE52815.2021.9662745.

Vojvodić, H., Vujić, M. and Škorput, P. (2017) 'Cooperative architecture of data acquisition for emission reduction in traffic', in *2017 25th Telecommunication Forum (TELFOR)*. *2017 25th Telecommunication Forum (TELFOR)*, pp. 1–4. Available at: https://doi.org/10.1109/TELFOR.2017.8249341.

Wang, P. *et al.* (2021) 'A two-stage method for bus passenger load prediction using automatic passenger counting data', *IET Intelligent Transport Systems*, 15(2), pp. 248–260. Available at: https://doi.org/10.1049/itr2.12018.

Wang, S. *et al.* (2021) 'Wi-Fi Sniffing Data Based Passenger-flow Detection Method Applied in Urban Rail Transit', *2021 IEEE 6th International Conference on Computer and Communication Systems, ICCCS 2021*, pp. 17–21. Available at: https://doi.org/10.1109/ICCCS52626.2021.9449240.

Wang, W. *et al.* (2024) 'Freight rate index forecasting with Prophet model based on multi-dimensional significant events', *Expert Systems with Applications*, 249, p. 123451. Available at: https://doi.org/10.1016/j.eswa.2024.123451.

Wei, L., Huang, L. and Huang, H. (2022) 'Research on Forecasting of Port Bulk Freight Volume during the COVID-19 Pandemic Based on Time Series Analysis', in *2022 International Conference on Management Engineering, Software Engineering and Service Sciences (ICMSS)*. *2022 International Conference on Management Engineering, Software Engineering and Service Sciences (ICMSS)*, pp. 132–140. Available at: https://doi.org/10.1109/ICMSS55574.2022.00028.

Westerman, M. (1995) *IVHS: Probe Vehicles*, *JPL's Wireless Communication Reference Website*. Available at: http://www.wirelesscommunication.nl/reference/chaptr01/roadtrin/ivhsprob.htm (Accessed: 27 September 2022).

Willmott, C.J. and Matsuura, K. (2005) 'Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance', *Climate Research*, 30(1), pp. 79–82.

Wu, W. *et al.* (2021) 'Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image', *PLoS ONE*, 16(10), p. e0259283. Available at: https://doi.org/10.1371/JOURNAL.PONE.0259283.

van Wyk, A. (2023) *Machine Learning with LightGBM and Python: A practitioner's guide to developing production-ready machine learning systems*. Packt Publishing. Available at: https://books.google.com.co/books?id=ArraEAAAQBAJ.

Yang, S., He, M. and Tao, Z. (2022) 'Research on theory and method of refinement clearing of urban rail transit passenger flow', in *Sixth International Conference on Electromechanical Control Technology and Transportation (ICECTT 2021)*. *Sixth International Conference on Electromechanical Control Technology and Transportation (ICECTT 2021)*, SPIE, pp. 545–552. Available at: https://doi.org/10.1117/12.2623918.

Ye, Y., Chen, L. and Xue, F. (2019) 'Passenger Flow Prediction in Bus Transportation System using ARIMA Models with Big Data', in *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 436–443. Available at: https://doi.org/10.1109/CyberC.2019.00081.

YOLOv5 | PyTorch (2022). Available at: https://pytorch.org/hub/ultralytics_yolov5/ (Accessed: 22 September 2022).

Yousuf, Md.S.I., Nijhum, S.M.R.H. and Khan, M.I. (2024) 'HybStock: A Hybrid Deep Learning-Based Model to Predict Stock Market on Dhaka Stock Exchange', in *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*. *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pp. 782–787. Available at: https://doi.org/10.1109/ICEEICT62016.2024.10534537.

Zaraket, K. *et al.* (2024) 'Hyper-Flophet: A neural Prophet-based model for traffic flow forecasting in transportation systems', *Simulation Modelling Practice and Theory*, 134, p. 102954. Available at: https://doi.org/10.1016/j.simpat.2024.102954.

Zhang, S. *et al.* (2020) 'Tiny YOLO Optimization Oriented Bus Passenger Object Detection', *Chinese Journal of Electronics*, 29(1), pp. 132–138. Available at: https://doi.org/10.1049/cje.2019.11.002.

Zhang, Y. *et al.* (2020) 'Bus passenger flow statistics algorithm based on deep learning', *Multimedia Tools and Applications*, 79(39–40), pp. 28785–28806. Available at: https://doi.org/10.1007/s11042-020-09487-0.

Zhao, D. *et al.* (2023) 'Trip purpose prediction using travel survey data with POI information via gradient boosting decision trees', *IET Intelligent Transport Systems*, n/a(n/a), pp. 1–21. Available at: https://doi.org/10.1049/itr2.12450.

Zhong, W., Jiang, Y.-Q. and Zhang, X. (2022) 'Research on Road Object Detection Algorithm Based on YOLOv5+Deepsort', in *2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*. *2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, pp. 644–648. Available at: https://doi.org/10.1109/ICICML57342.2022.10009649.

Zhou, Z.-H. (2012) *Ensemble Methods: Foundations and Algorithms*. New York: Chapman and Hall/CRC. Available at: https://doi.org/10.1201/b12207.

Zhu, F. *et al.* (2008) 'Research on Counting Method of Bus Passenger Flow Based on Kinematics of Human Body', in *2008 International Conference on Computer Science and Software Engineering*. *2008 International Conference on Computer Science and Software Engineering*, pp. 201–204. Available at: https://doi.org/10.1109/CSSE.2008.611.

Zhu, L. *et al.* (2018) 'Urban link travel time estimation using traffic states-based data fusion', *IET Intelligent Transport Systems*, 12(7), pp. 651–663. Available at: https://doi.org/10.1049/iet-its.2017.0116.

Zhu, W., Zhang, S. and Liang, J. (2023) 'Model Construction and Analysis of Bus User Portrait Based on Multivariate Data Fusion', in *2023 Asia-Europe Conference on*

Electronics, Data Processing and Informatics (ACEDPI). *2023 Asia-Europe Conference on Electronics, Data Processing and Informatics (ACEDPI)*, pp. 194–202. Available at: https://doi.org/10.1109/ACEDPI58926.2023.00045.