**UNIVERISTY OF PIRAEUS - DEPARTMENT OF INFORMATICS**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**MSc Cybersecurity and Data Science**

ΠΜΣ «Κυβερνοασφάλεια και Επιστήμη Δεδομένων»

**MSc Thesis**

Μεταπτυχιακή Διατριβή

| | |
|---|---|
| **Thesis Title:**<br>Τίτλος Διατριβής: | **Urban Shared Mobility Demand Forecasting**<br>Πρόβλεψη Ζήτησης για Αστική Κινητικότητα |
| **Student's name-surname:**<br>Ονοματεπώνυμο φοιτητή: | **Antonios Tziorvas**<br>Αντώνιος Τζιόρβας |
| **Father's name:**<br>Πατρώνυμο: | **Ioannis**<br>Ιωάννης |
| **Student's ID No:**<br>Αριθμός Μητρώου: | ΜΠΚΕΔ 2342 |
| **Supervisor:**<br>Επιβλέπων: | **Ioannis Theodoridis, Professor**<br>Ιωάννης Θεοδωρίδης, Καθηγητής |

June 2025 / Ιούνιος 2025

**3-Member Examination Committee**

Τριμελής Εξεταστική Επιτροπή

**Yannis Theodoridis**
**Professor**

Ιωάννης Θεοδωρίδης
Καθηγητής

**Nikos Pelekis**
**Professor**

Νικόλαος Πελέκης
Καθηγητής

**Aggelos Pikrakis**
**Assoc. Professor**

Άγγελος Πικράκης
Αναπληρωτής Καθηγητής

# ABSTRACT

Urban demand forecasting plays a critical role in optimizing routing, dispatching, and congestion management within Intelligent Transportation Systems. By leveraging data fusion and analytics techniques, traffic density estimation serves as a key intermediate measure for identifying and predicting emerging demand patterns. In this thesis, we propose two gradient boosting model variations, one for classification and one for regression, both capable of generating demand forecasts at various temporal horizons, from 5 minutes up to one hour. Our approach effectively integrates spatial and temporal features, enabling accurate predictions that are essential for improving the efficiency of shared (micro-)mobility services. To evaluate the effectiveness of our approach, we utilize open shared mobility data derived from e-scooters and e-bikes networks in two Dutch metropolitan areas. These real-world datasets enable us to validate our approach and demonstrate its effectiveness in capturing the complexities of modern urban mobility. Ultimately, our methodology offers novel insights on urban micro-mobility management, helping to tackle the challenges arising from rapid urbanization and thus, contributing to more sustainable, efficient, and livable cities.

Η πρόβλεψη της αστικής ζήτησης διαδραματίζει κρίσιμο ρόλο στη βελτιστοποίηση της δρομολόγησης, της αποστολής και της διαχείρισης της κυκλοφοριακής συμφόρησης στα Έξυπνα Συστήματα Μεταφορών. Αξιοποιώντας τεχνικές συγχώνευσης και ανάλυσης δεδομένων, η εκτίμηση της κυκλοφοριακής πυκνότητας χρησιμεύει ως βασικό ενδιάμεσο μέτρο για τον εντοπισμό και την πρόβλεψη αναδυόμενων προτύπων ζήτησης. Σε αυτή τη διατριβή, προτείνουμε δύο παραλλαγές του μοντέλου gradient boosting, μία για ταξινόμηση και μία για παλινδρόμηση, και οι δύο ικανές να παράγουν προβλέψεις ζήτησης σε διάφορους χρονικούς ορίζοντες, από 5 λεπτά έως μία ώρα. Η προσέγγισή μας ενσωματώνει αποτελεσματικά χωρικά και χρονικά χαρακτηριστικά, επιτρέποντας ακριβείς προβλέψεις που είναι απαραίτητες για τη βελτίωση της αποτελεσματικότητας των υπηρεσιών της διαμοιραζόμενης (μικρο-)κινητικότητας. Για να αξιολογήσουμε την αποτελεσματικότητα της προσέγγισής μας, χρησιμοποιούμε ανοιχτά δεδομένα κοινής κινητικότητας που προέρχονται από δίκτυα ηλεκτρικών σκούτερ και ηλεκτρικών ποδηλάτων σε δύο μητροπολιτικές περιοχές της Ολλανδίας. Αυτά τα πραγματικά δεδομένα μας επιτρέπουν να επικυρώσουμε την προσέγγισή μας και να αποδείξουμε την αποτελεσματικότητά της στην καταγραφή των πολυπλοκοτήτων της σύγχρονης αστικής κινητικότητας. Εν τέλει, η μεθοδολογία μας προσφέρει νέες πληροφορίες σχετικά με τη διαχείριση της αστικής μικροκινητικότητας, βοηθώντας στην αντιμετώπιση των προκλήσεων που προκύπτουν από την ταχεία αστικοποίηση και συμβάλλοντας έτσι σε πιο βιώσιμες, αποδοτικές και κατοικήσιμες πόλεις.

# 1 INTRODUCTION

The significance of time series forecasting [41] has grown considerably in recent years, primarily due to the exponentially increasing availability of sequential data across diverse domains. Notable applications include are finance and marketing (e.g., stock price forecasting and analysis of historical sales), medicine (e.g., electroencephalography and monitoring of heart rate), energy management (e.g., forecasting electricity and water consumption), and meteorology [1, 18] (rainfall and temperature prediction).This growing interest reflects the escalating complexity of systems in the real world, where accurate and timely forecasts are critical to informed decision-making and efficient allocation of resources.

As time series analysis has advanced, it has evolved to encompass spatio-temporal (ST) models—approaches that integrate both spatial and temporal dimensions. These models are especially well-suited to modern problems that involve geographically dispersed systems, such as domains of climate science (e.g., long-term weather prediction and climate change simulation) and urban data analytics, such as traffic flow prediction, demand forecasting for shared mobility, and energy consumption profiling for urban planning. The spatio-temporal paradigm addresses the limitations of solely temporal models by incorporating spatial dependencies, thus providing a more profound view of dynamic systems that change over both spatial and temporal dimensions.

Among these applications, urban shared mobility systems—such as bike-sharing, e-scooters, and ride-pooling platforms—have emerged as an example of such systems [42]. These services generate rich, high-frequency data streams that capture human mobility patterns in real-time, influenced factors such as time of day, geographic topology, weather conditions, and social events. The ability to accurately forecast short-term demand [13] in these systems is critical for a wide range of stakeholders: from operators seeking to balance vehicle distribution and reduce user wait times, to city planners aiming to support sustainable transport policies and infrastructure investments.

Despite advancements in modeling techniques, forecasting demand in urban shared mobility systems remains a challenging task. Many existing approaches fail to fully capture the nonlinear and spatially heterogeneous patterns inherent in these systems [14], especially under constraints of limited interpretability and model generalizability across cities or modalities. This thesis investigates these challenges through a data-driven lens, exploring and benchmarking various forecasting strategies—including classical, machine learning, and deep learning models—focused on their ability to model spatio-temporal dynamics in shared mobility demand.

## 1.1 Background and Motivation

Urban mobility systems are undergoing a paradigm shift, catalyzed by the emergence and rapid proliferation of Mobility-as-a-Service [13] (MaaS) platforms. Urban Shared Mobility—an essential subset of MaaS—comprises a wide array of transportation solutions that are either concurrently (i.e., shared simultaneously among multiple users) or sequentially (i.e., used one after another) accessed by the public. These include public transportation, micro-mobility services (e.g., bike- and scooter-sharing), commute-based models (e.g., carpooling), and on-demand services (e.g., ride-hailing and ridesharing) [34]. Figure 1-1 illustrates the diversity and integration of these services within the broader MaaS ecosystem.

This evolution reflects broader socio-economic shifts: increasing urbanization, sustainability mandates, and the growing demand for flexible, on-demand transportation. As cities transition toward smarter infrastructures, shared mobility systems are emerging as crucial components of multi-modal and user-centric transportation networks. However, these systems are inherently complex, dynamic, and context-sensitive. Demand varies drastically across time and space, driven by factors (time-of-day and day-of-week cyclicality, weather fluctuations, population density and land use, urban infrastructure and public transport integration social events and economic activity).

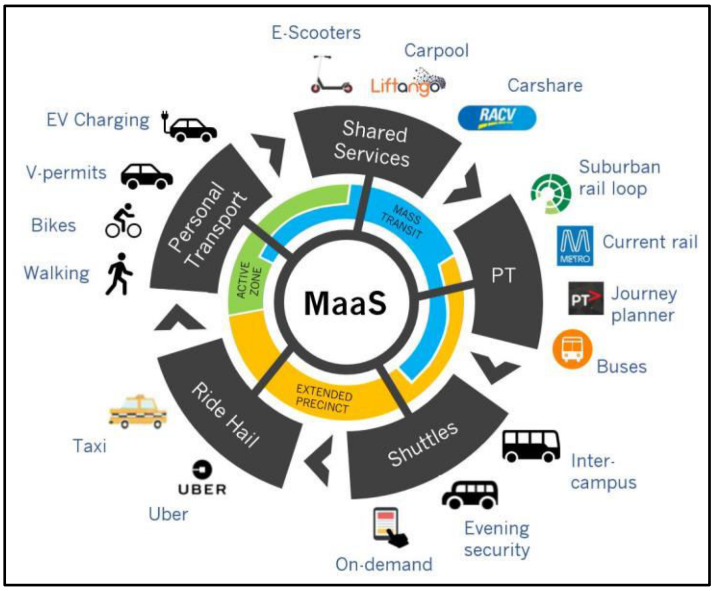


**Figure 1-1: Mobility-as-a-Service Ecosystem**
**(source: https://www.monash.edu/campus-sustainability/sustainability-strategy/Transport)**

Accurate demand forecasting thus becomes an indispensable operational tool. It enables service providers to anticipate mobility needs, optimize vehicle distribution, improve fleet utilization, and reduce user wait times. From a policy and governance perspective, forecasting supports long-term infrastructure planning, public transportation expansion, and emergency response strategies. Beyond shared mobility, similar forecasting challenges emerge in traffic flow prediction and Intelligent Transportation Systems (ITS). Here, the real-time analysis of spatio-temporal patterns helps manage congestion, reduce travel times, and enhance road safety. Spatio-temporal models, which integrate both temporal sequences and spatial correlations, are increasingly leveraged in these contexts for their ability to represent the intertwined behavior of mobility demand and urban dynamics.

The motivation behind this thesis is to explore, evaluate, and enhance the current state of forecasting within the Urban Shared Mobility domain. By identifying and addressing existing bottlenecks and limitations in prediction methodologies, this research aims to contribute to the development of more intelligent and sustainable transportation systems. Improved forecasting not only elevates user satisfaction and service responsiveness but also supports strategic resource deployment—fostering more sustainable, equitable, and resilient urban environments.

## 1.2 Current Limitations

With the proliferation of sensor networks and the increasing availability of high-frequency data, the complexity of real-world time series forecasting has evolved substantially. Contemporary datasets frequently incorporate multiple modalities, with the spatial dimension emerging as particularly prominent in domains such as traffic monitoring, climate modeling, and urban mobility. These spatial features introduce dependencies and interactions that traditional time series models, which rely solely on temporal correlations, are fundamentally unequipped to model. As a result, classical methods such as ARIMA or even standard LSTM-based models suffer from reduced predictive performance in applications where location-based interactions are central.

In response to these limitations, a range of modeling strategies have been developed to incorporate spatial dependencies into temporal forecasting. These include spatially-aware Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) [3] variants, and, more recently, Graph Neural Networks (GNNs) [5] and diffusion-based architectures. While these models have demonstrated notable improvements in capturing spatio-temporal dynamics, they also introduce substantial challenges related to scalability, generalization, and computational efficiency.

The advent of Spatio-Temporal Graphs (STGs) has brought further attention to these issues. In STGs, nodes typically represent spatial entities such as intersections, bike stations, or city zones, while edges encode functional relationships over time—such as shared mobility flow, road connectivity, or user co-location. These graphs are often dynamic, non-Euclidean, and large-scale, posing substantial challenges for sequence-based models like RNNs and standard convolutional neural networks. To address these limitations, Spatio-Temporal Graph Neural Networks [15] (STGNNs) and their convolutional variants (STGCNs) have been proposed. These methods jointly learn temporal dependencies and spatial graph structures, achieving improved forecasting performance in complex urban systems. However, their architectural sophistication comes at the cost of significantly increased model complexity, higher computational requirements, and greater susceptibility to overfitting in data-sparse regimes.

Even when training complexity is successfully managed, inference-time performance remains a critical bottleneck, particularly in latency-sensitive applications such as real-time traffic prediction, dynamic vehicle rebalancing, and emergency response systems. Models that rely on diffusion-based representations, which simulate stochastic signal propagation across graph structures, tend to be especially resource-intensive at inference. While post-training optimization techniques—such as model pruning, quantization [16], or knowledge distillation [17]—have been adopted to reduce computational load, they frequently entail a trade-off between model speed and predictive accuracy.

These limitations collectively highlight the gap between theoretical advancements in spatio-temporal modeling and their real-world deployability in urban mobility systems. Bridging this gap is a central challenge that motivates the development of more efficient, interpretable, and operationally viable forecasting models—an objective that this thesis aims to address.

## 1.3 Contribution

In this thesis, we provide a robust feature extraction pipeline that can capture both spatial and temporal dependencies and integrate them into our model. We also present a gradient boosting ML algorithm capable of predicting the demand in a given area, either in the form of levels (e.g., 'Low', 'Medium', 'High') through a classifier, or absolute demand values via a regressor. The performance of our model is evaluated using real-world micro-mobility data and the results seem quite promising since our approach can effectively adapt to the uniqueness of each area and adequately model its intricacies.

The rest of this thesis is structured as follows: Section 2 reviews upon the related work, Section 3 introduces our forecasting methodology, including the feature extraction process and training configuration. Section 4 explains the experimental setup and presents the results of both predictive models, offering detailed performance metrics and comparative analysis across different districts and forecasting horizons. Finally, Section 5 concludes the thesis with a summary of the key contributions and directions for future research.

# 2 LITERATURE REVIEW

Spatio-temporal forecasting has garnered increasing attention in recent years, primarily due to the rapid growth in the availability of high-resolution datasets that simultaneously capture both spatial and temporal dynamics. These datasets emerge from a range of domains—including transportation networks, environmental monitoring systems, and urban sensor grids—where understanding the evolution of patterns over space and time is critical. Among the most prominent application areas is traffic flow prediction, which serves as a benchmark task for evaluating spatio-temporal models due to its complexity and real-time constraints [16, 17, 19].

A wide variety of modeling approaches have been proposed to address the challenges associated with spatio-temporal forecasting. These range from classical statistical methods, such as Autoregressive Integrated Moving Average (ARIMA) models, to machine learning techniques like decision trees and random forests, and more recently, to deep learning architectures that leverage the representational power of recurrent, convolutional, and graph-based networks. In parallel, advances in foundation models have introduced pre-trained, general-purpose models capable of capturing long-range temporal dependencies and spatial structures through transfer learning and self-supervised training.

In this section, we review the main categories of spatio-temporal forecasting approaches relevant to urban mobility and demand prediction. First, we examine classical machine learning techniques, focusing on their strengths and limitations in modeling temporal patterns and spatial aggregation. Next, we explore deep learning-based architectures (including Graph Neural Networks (GNNs), Spatio-Temporal Graph Convolutional Networks (STGCNs), and diffusion-based models), which have become the state-of-the-art in recent years. Finally, we discuss the emerging class of foundation models, which aim to generalize forecasting across domains by leveraging large-scale temporal pretraining.

## 2.1 Foundations of Time Series Forecasting

Before the integration of spatial information into forecasting models, traditional statistical techniques dominated the field of temporal prediction. Among these, Autoregressive Integrated Moving Average (ARIMA) models and their variants (SARIMA), along with Kalman filters, have been widely applied to domains like mobility demand, traffic flow prediction, and resource utilization. These models operate on the assumption of stationarity, linearity, and temporal autocorrelation, making them effective for capturing predictable, short-term trends in time series data. However, they encounter limitations when tasked with modeling nonlinear dynamics, handling long-range dependencies, or incorporating complex contextual factors beyond the temporal dimension. As a result, while they remain useful for relatively simple forecasting tasks, these methods fall short in environments characterized by complex, dynamic patterns—such as urban traffic or mobility systems—where spatial dependencies and external influences are critical.

Machine Learning (ML) approaches such as Decision Trees, Support Vector Regression (SVR), and ensemble techniques like Random Forests and Gradient Boosting Machines (e.g., XGBoost) introduced nonlinearity and robustness to noise. These models typically require substantial manual feature engineering and still treat spatial dependencies indirectly—if at all. Nonetheless, they served as crucial stepping stones toward more complex architectures and remain competitive baselines for spatio-temporal tasks.

### 2.1.1 Statistical Models

Kalman Filter-based approaches gained traction due to their recursive parameter estimation and adaptability to streaming data. Guo et al. (2014) [30] proposed an Adaptive Kalman Filter (AKF) architecture for short-term stochastic traffic forecasting and uncertainty quantification. Their approach is built upon a SARIMA + GARCH structure that captures both the conditional mean and variance of 15-minute interval traffic flow data. The SARIMA component accounts for seasonal patterns in traffic flow, while the GARCH component models the volatility, allowing the system to react to sudden fluctuations in traffic. This hybrid structure is translated into a state-space representation, enabling real-time inference through Kalman filtering. Crucially, the authors adopt the Myers and Tapley (1976) adaptation mechanism to dynamically update the state and observation process covariance matrices based on historical estimation errors. This adaptive mechanism allows the AKF to track changes in traffic variance more effectively than standard Kalman filters, which assume static process noise. The filter recursively estimates both point forecasts and prediction intervals through sequential seasonal exponential smoothing, coupled with two adaptive Kalman filters—one for the ARMA process and another for the GARCH component. Empirical evaluations on datasets from the UK and US demonstrate the superiority of the AKF approach against baseline models, and a conventional Kalman Filter with fixed variances. Results show that the AKF not only improves forecasting accuracy during high volatility periods but also offers robust online updates, making it particularly suitable for real-time traffic prediction systems.

Duan et al. (2018) [11] introduced the unified STARIMA (*uSTARIMA*) model to improve short-term road traffic forecasting by addressing a key limitation in traditional spatio-temporal models: the inability to account for the time-varying nature of spatial correlations across traffic measurement points. These correlations, which fluctuate due to changes in road network topology, average speed, and trip distribution throughout the day, are inadequately captured by static-lag models. uSTARIMA addresses this by incorporating time-varying lag structures into a single unified model, removing the need to calibrate separate models for different time intervals. The proposed methodology begins with standard preprocessing steps, including noise filtering, missing data handling, and weekday/weekend classification. To model traffic dynamics effectively, the authors divide the day into "peak" and "off-peak" intervals using the *ISODATA* clustering algorithm, which groups time periods based on observed flow and speed. The core innovation lies in estimating the lag between detector stations as a function of travel time, calculated from distance and estimated spatial mean speed (SMS) - a metric more representative of travel conditions than the detector-measured temporal mean speed (TMS). Finally, the traditional STARIMA formulation is retained, but with a modified spatial weight matrix Γ, whose elements incorporate the computed time-varying lags. This allows the model to dynamically adjust the temporal alignment of spatial inputs, enabling more accurate and context-sensitive forecasting without overcomplicating the overall structure.

### 2.1.2 Classical Machine Learning

Feng et al. (2021) [35] developed a predictive framework for bike-sharing demand in Chicago, leveraging Poisson regression to account for count-based demand data. Their model incorporated a diverse set of features- including temporal variables, meteorological conditions, population density, and activity density - to capture the multifactorial nature of bike usage. To improve performance, the study integrated a Random Forest ensemble, which enhanced predictive accuracy by capturing nonlinear relationships and feature interactions. This hybrid approach not only facilitated more precise demand-supply alignment but also aided in strategic decision-making, such as the placement of new bike stations. In a related study, Xiao et al. [36] focused on short-term car-hailing demand prediction in Hefei, China, employing clustering techniques to uncover spatial-temporal demand patterns. They proposed an improved k-nearest neighbors (k-NN) model, enhanced by prior clustering of demand data, which enabled real-time forecasting for dynamic ride-hailing platforms. Their findings underscore the utility of combining unsupervised learning with traditional regression/classification models to improve prediction granularity and responsiveness in mobility systems.

An innovative approach to bike-share station traffic prediction is the two-step pattern recognition framework introduced by Sohrabi and Ermagun [37], which integrates spatiotemporal pattern recognition with machine learning techniques. In the first stage, the method constructs station-level traffic profiles by segmenting bike-sharing activity into time series sequences. These sequences are divided into intervals of $t$ minutes, with an overlap of length $O$ between consecutive profiles. This temporal segmentation captures the dynamic usage patterns of bikes while ensuring continuity across the observation windows. In the second stage, a k-nearest neighbors (kNN) algorithm is employed to identify spatiotemporal similarities among the generated profiles. The similarity search utilizes a weighted Euclidean distance metric that incorporates historical traffic volumes along with auxiliary temporal (e.g., weather conditions, day of the week) and spatial (e.g., socio-demographic factors, land use, transportation infrastructure) features. This combined modeling approach facilitates both short-term and long-term traffic forecasting, applied to both individual stations and system-wide scales. The results underscore the effectiveness of integrating temporal segmentation with spatially informed feature engineering, highlighting the potential of hybrid models for demand forecasting in urban mobility systems.

A large-scale machine learning framework for predicting non-motorized traffic patterns was developed by Smith et al. [38], with a focus on bicycle and pedestrian traffic across 20 major U.S. metropolitan statistical areas (MSAs). Building on the work of Le et al. [39], who utilized stepwise linear regression models informed by Census-derived, neighborhood-level covariates, Smith et al. enhanced predictive performance by incorporating high-resolution, street-level features. Their dataset encompassed traffic counts from 4,145 distinct observation points, enriched with Google Street View imagery and Point of Interest (PoI) data, thereby providing a comprehensive understanding of the urban context. The authors evaluated a variety of machine learning algorithms—including linear regression, ridge regression, lasso, bagging, gradient boosting, and random forest—demonstrating that models trained on this augmented multimodal dataset substantially outperformed traditional linear approaches. These findings underscore the importance of spatially granular data and ensemble-based learning techniques in improving the accuracy of non-motorized traffic demand forecasting, especially in diverse and heterogeneous urban environments.

## 2.2 Deep Learning for Temporal Modeling

Deep learning has revolutionized time series forecasting by enabling models to capture complex, non-linear relationships and long-range dependencies, which traditional methods often struggle to model. Among the most prominent architectures for temporal forecasting are Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs). While traditional RNNs suffer from issues like vanishing gradients, LSTMs introduce memory cells that help mitigate this problem, allowing them to retain long-range dependencies in the data. GRUs, a simplified version of LSTMs, offer similar benefits while reducing computational complexity. Nevertheless, these architectures excel at forecasting tasks where patterns emerge over time and need to be learned from past data (including urban mobility forecasting, traffic flow prediction, ride-hailing demand, and shared mobility systems),by learning temporal patterns from sequential data. The advantage of LSTMs and GRUs lies in their ability to model complex temporal patterns and incorporate external features (e.g., weather, time-of-day, or holidays), making them effective for dynamic environments like traffic forecasting and bike-sharing demand.

The introduction of the Transformer architecture (Vaswani et al., 2017) [22] represented a paradigm shift in the field of sequence modeling. By replacing recurrence with a self-attention mechanism, Transformers enabled the parallel processing of sequences and allowed models to learn long-range dependencies more efficiently. This innovation significantly improved the modeling of temporal dynamics, especially in settings where context from distant time steps is essential. While originally developed for natural language processing, the Transformer architecture quickly gained traction in the time series domain due to its scalability and flexibility. Its ability to assign varying importance to different time steps has proven particularly useful in spatio-temporal contexts, where temporal patterns often interact with complex spatial structures. Subsequent adaptations of Transformers have formed the foundation for a new generation of spatio-temporal forecasting models.

### 2.2.1 LSTMs and CNNs

Instead of relying on shallow models and manually crafted features derived from physical network structures. Lv et al. (2014) [12] proposed a deep learning-based framework that leverages the growing availability of traffic big data to learn spatio-temporal patterns directly from raw observations. Rather than depending on explicitly defined macroscopic models or handcrafted features, their approach introduces a Stacked Autoencoder (SAE) architecture designed to autonomously capture the underlying structure of traffic flow data. The SAE model is constructed by sequentially stacking multiple autoencoders—neural networks trained to reproduce their input—to progressively extract higher-level abstractions of the input data. This deep representation learning is achieved through greedy layer-wise pretraining, followed by fine-tuning using supervised backpropagation. A logistic regression layer is attached to the final encoder output to perform the actual traffic flow prediction. By incorporating data across multiple freeway locations (spatial dimension) and historical time intervals (temporal dimension), the model implicitly captures the intricate dependencies in traffic flow without manual intervention. Empirical results demonstrate that optimizing the number of hidden layers and the size of each layer significantly enhances predictive performance, showcasing the potential of deep architectures in complex, data-rich forecasting environments.

Despite the emergence of increasingly sophisticated traffic forecasting architectures, a substantial portion of recent work continues to rely on the foundational principles of combined modeling using Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs). Zhang et al. [31] proposed a short-term urban traffic flow prediction framework that exemplifies this tendency. Their approach integrates CNNs, to capture localized spatial patterns across an urban grid, with LSTMs, to model sequential temporal dependencies inherent in traffic flow data. By aligning spatial inflow/outflow dynamics with temporal trends, proximities, and periodicities, the model offers a more comprehensive view of traffic behavior than purely temporal or purely spatial approaches. However, it is important to note that this framework—like many contemporary methods—remains grounded in earlier deep learning strategies and does not fundamentally rethink the underlying representation of spatio-temporal correlations. Instead, it refines traditional CNN-LSTM combinations through better data fusion techniques and modular architectures. While effective, particularly when external factors such as weather or special events are incorporated, these models inherit limitations from their component architectures, including constrained scalability and difficulties in capturing complex relational structures beyond grid-based approximations. This ongoing reliance on classical deep learning building blocks underscores the need for more expressive and flexible approaches capable of natively modeling the irregular and dynamic nature of urban mobility networks.

In an effort to improve the scalability and accuracy of multi-step traffic forecasting, Essien et al. [21] introduced a ConvLSTM-based framework that rethinks how traffic data is represented and processed. Rather than treating time series data directly, their approach encodes spatiotemporal traffic patterns into image-like structures using recurrence quantification analysis. This transformation allows the model to capture recurring traffic states in a structured spatial format, effectively bridging temporal evolution with spatial topology. The encoded representations are then processed using a ConvLSTM architecture, which simultaneously performs spatial feature extraction and temporal sequence modeling. This approach marks a notable shift from traditional sequence-based models by leveraging visual encoding to preserve complex spatial dependencies across an urban network. Unlike earlier CNN-LSTM hybrids that rely on grid-based approximations or handcrafted features, this model demonstrates how time series visualization can be harnessed for deep learning, offering both robustness and scalability. However, despite this conceptual advancement, the model still inherits certain constraints from its sequential backbone, particularly with respect to long-range dependencies and graph-structured spatial relations—limitations that more recent graph-based models seek to overcome.

### 2.2.2 The Transformer Era

Essien et al. (2020) [20] proposed the Self-Attention Convolutional LSTM (SA-ConvLSTM) model to improve the accuracy of spatiotemporal demand forecasting in online car-hailing services. The authors argue that accurate predictions are essential for enhancing user experience, optimizing operational efficiency, and supporting urban mobility planning. They identify key limitations in earlier approaches, including excessive granularity in grid-based image representations that lead to data sparsity, and the underutilization of recent advances in deep learning, particularly mechanisms capable of modeling long-range dependencies. To address these issues, the SA-ConvLSTM integrates a self-attention mechanism within each ConvLSTM layer, allowing the model to capture global spatial correlations by computing pairwise similarity scores across all spatial positions. The architecture also includes additional memory units that enhance the model's capacity to retain long-term spatiotemporal patterns. The methodology involves converting online car-hailing trajectories into grid-aligned image sequences, followed by data augmentation through image cropping to address sparsity and irregularities. The model was evaluated on a real-world dataset in Chengdu and benchmarked against various traditional and deep learning baselines, including LSTM, GRU, CNN, and standard ConvLSTM models. Experimental results demonstrate that SA-ConvLSTM consistently outperforms these baselines across multiple evaluation metrics, including MAE, RMSE, SSIM, and $R^2$. The study further concludes that encoding spatiotemporal information in image form yields superior forecasting performance compared to representing spatial data as sequential features, underscoring the effectiveness of the proposed approach in capturing complex spatiotemporal dependencies in mobility demand data.

Zheng Chuanpan et al. (2019) [6] introduced the Graph Multi-Attention Network (GMAN), a hybrid architecture that combines the power of graph-based representations with attention mechanisms to address the challenges of long-term traffic prediction. GMAN is designed to model the complex spatial and temporal dependencies in traffic data by treating road networks as graphs, where intersections or road segments are represented as nodes, and traffic flow is captured through edges. This graph-based representation is enhanced by multi-head attention mechanisms, which enable the model to focus on both local and global dependencies in the traffic network. GMAN employs a dual attention mechanism: the first attention layer learns the temporal correlations at each node (road segment), while the second attention layer captures the spatial dependencies across the network. This ability to capture both spatial and temporal patterns make GMAN particularly effective for long-term traffic prediction, where both types of dependencies are crucial. The model demonstrated superior performance on hourly prediction tasks, outperforming traditional models and exhibiting robustness across multiple traffic datasets.

In a similar vein, Zheng Haifeng et al. (2023) [23] introduced a novel attention-based Conv-LSTM module to efficiently extract spatial-temporal features, enabling the simultaneous processing and weighting of traffic flow sequences. Unlike hybrid models that process spatial and temporal features separately, this approach integrates these features directly within the Conv-LSTM module. The attention mechanism autonomously assigns weights to different time steps, focusing on the most relevant historical information without needing external data like road speed. In addition, two Bi-LSTM modules capture the periodicity of traffic flow, accounting for both daily and weekly patterns by considering both forward and backward sequences. The spatial-temporal features from the Conv-LSTM and periodic features from the Bi-LSTMs are then fused in a feature fusion layer, providing a comprehensive representation of traffic flow dynamics. The final forecasting is performed through two fully connected layers. This integrated approach offers a significant advantage over previous models, which often process spatial, temporal, and periodic features independently, and experimental results demonstrate its superior predictive performance.

## 2.3 Modeling Spatial Dependencies

The introduction of Spatio-Temporal Graphs (STGs) marked a pivotal advancement in the modeling of complex spatio-temporal phenomena. Unlike traditional time series models that treat spatial and temporal dimensions independently or sequentially, STGs provide a unified framework that explicitly captures both spatial dependencies among entities (e.g., road segments, stations, regions) and temporal dynamics over time. This dual representation significantly improves forecasting accuracy, especially in domains such as traffic flow prediction, urban mobility analysis, and environmental monitoring, where spatial interactions are non-trivial and evolve dynamically over time.

This graph-based abstraction is particularly powerful in dynamic and heterogeneous environments, as highlighted by Yao et al. (2017) [40]. Their work demonstrates how entity graphs can encode rich spatio-temporal interactions among connected devices, a concept that directly informs how STGs capture mobility demand and traffic patterns. By structuring data in this way, STGs enable fine-grained learning of both spatial topology and temporal trends. Temporal patterns are modeled either as discrete snapshots (graph sequences) or continuous time-evolving processes. Crucially, the use of Graph Convolutional Networks (GCNs) and their spatio-temporal extensions, such as Spatio-Temporal Graph Convolutional Networks (STGCNs) [7] and Graph-Attention Networks [8], allows for localized and learnable information aggregation across spatially connected nodes while preserving temporal ordering. These models outperform earlier CNN or RNN-based approaches by learning structural relationships instead of relying solely on grid-based approximations. In parallel, hypergraphs have emerged as an enhancement to classical graph representations by allowing edges (hyperedges) to connect more than two nodes.

This is particularly beneficial for encoding complex higher-order spatial interactions, such as multi-way relationships in mobility demand (e.g., shared mobility hubs serving multiple neighborhoods). The integration of such representations has further enriched the expressiveness of spatio-temporal models.

Overall, STGs have become a cornerstone in the evolution of spatio-temporal forecasting architectures. They not only bridge the gap between spatial topology and temporal dynamics but also provide a scalable and interpretable structure for modeling real-world mobility systems. Their incorporation into hybrid architectures (e.g., GCN-LSTM, STGCN, Diffusion Convolutional Recurrent Networks) reflects their central role in driving performance improvements and interpretability in urban forecasting tasks.

### 2.3.1 Spatio-Temporal Graph Neural Networks (STGNNs)

Park Cheonbok et al. (2020) [19] proposed ST-GRAT, a Spatio-Temporal Graph Attention Network specifically designed for dynamic road speed forecasting. The model incorporates multi-head attention mechanisms to adaptively weigh the influence of neighboring nodes based on evolving traffic conditions. This allows ST-GRAT to capture diverse spatial and temporal patterns more effectively. The architecture explicitly models spatio-temporal dependencies using separate spatial and temporal attention modules, enabling a more nuanced understanding of the complex interactions and temporal evolutions inherent in traffic systems. Additionally, the introduction of sentinel vectors equips the model with the ability to selectively attend to relevant neighbor information or rely on its own internal representations, thereby enhancing robustness and reducing susceptibility to noisy or irrelevant inputs. Together, these components allow ST-GRAT to achieve state-of-the-art performance in traffic speed prediction, particularly under dynamically changing conditions.

The next year Ali Ahmad et al. (2021) [27] proposed GCN-DHSTNet, a model that was able to make predictions of traffic flows considering external factors such as weather conditions. This model utilizes an unweighted graph to capture the topological structure of road segments. Each road segment is treated as a node in the graph, with the connections between segments represented as edges. The GCN component operates on a graph representation of the road network, where each node represents a spatial region and edges capture inter-regional relationships. This enables the model to learn both local and global spatial correlations. Temporal patterns are modeled using three parallel LSTM-based branches: recent, daily, and weekly, each designed to capture traffic dynamics at different temporal granularities. The recent branch focuses on immediate past conditions, the daily branch captures recurring intraday trends, and the weekly branch models periodic weekly behaviors. These temporal features are then fused using a parametric tensor-based mechanism, which learns to weigh each branch’s contribution to the final prediction. This multi-branch design allows GCN-DHSTNet to effectively model complex spatio-temporal interactions in traffic data.

Li et al. (2022) [28] introduced a Spatio-Temporal Graph Neural Network (STGNN) designed specifically for traffic flow prediction. The core contribution of their work lies in the integration of a Spatio-Temporal Graph (STG), which enables the model to jointly capture spatial and temporal dependencies inherent in traffic networks. Spatial dependencies, such as the influence of neighboring road segments, and temporal dependencies, such as recurring traffic patterns, are learned simultaneously through a unified graph-based architecture. In contrast to models that treat spatial and temporal correlations independently or through sequential layers, their approach fuses both aspects concurrently using graph structures. This design allows for a more expressive representation of complex interactions within the data. The road network is modeled as a directed graph, where nodes correspond to road segments and edges reflect directional traffic influence. This directionality is critical for modeling propagation effects within dynamic traffic systems. To efficiently learn node embeddings, the model incorporates neighborhood aggregation mechanisms inspired by GraphSAGE. Rather than learning fixed embeddings directly, the network is trained to learn an embedding function that generalizes across nodes based on their attributes (e.g., textual descriptions, degrees, or other features). This approach promotes better generalization and scalability in large, dynamic urban traffic networks.

Luo et al. (2023) [29] proposed GT-LSTM, a novel architecture for traffic prediction that leverages a self-adaptive graph structure to dynamically learn relationships between traffic nodes without relying on predefined spatial graphs (e.g., based on distance or similarity metrics). This flexibility allows the model to better reflect the evolving nature of urban traffic systems. The model consists of two core components: Feature Splicing and Pattern Capturing. In the Feature Splicing module, spatial dependencies are first captured independently using a self-adaptive Graph Convolutional Network (GCN) and then concatenated with raw temporal features. This non-interleaved (non-inter-embedded) design contrasts with conventional approaches that merge spatio-temporal information within a single block. Instead, GT-LSTM defers fusion until after spatial extraction, thereby preserving the integrity of temporal dynamics during early processing. The concatenated features are then passed into the Pattern Capturing module, which includes a Temporal Convolutional Network (TCN) and a Bidirectional Long Short-Term Memory (Bi-LSTM) network. This two-stage design allows the model to effectively capture long-range temporal dependencies while maintaining spatial context. By decoupling spatial and temporal modeling phases, the GT-LSTM achieves greater flexibility and improves the representation capacity of spatio-temporal sequences.

### 2.3.2 Diffusion-Based STG Architectures

While traditional Spatio-Temporal Graph Neural Networks (STGNNs) successfully integrate graph structures with temporal modeling, they often rely on deterministic message-passing schemes that limit their ability to capture the inherent uncertainty and variability of dynamic urban systems. In response to these limitations, diffusion-based architectures have emerged, introducing stochastic modeling mechanisms that better represent the complex propagation of information across spatio-temporal graphs. Denoising-diffusion based approaches have gained much popularity [4] in the last 2 years with almost over half of the released papers regarding timeseries forecasting are based on these. Diffusion models treat the flow of information as a probabilistic diffusion process over the graph, enabling them to capture not only local spatial dependencies but also long-range interactions with variable strength and temporal delays.

One of the early advances in this direction is the Diffusion Convolutional Recurrent Neural Network (DCRNN) (Li et al., 2017) [2], which frames traffic forecasting as a sequence-to-sequence learning task over spatio-temporal graphs. In DCRNN, spatial dependencies are captured via a diffusion convolution operation that simulates bidirectional random walks on a graph, allowing traffic information to propagate probabilistically across the network. This is combined with a gated recurrent unit (GRU) to model temporal dynamics, enabling the network to learn complex, nonlinear spatio-temporal correlations. DCRNN demonstrated superior performance over traditional GCN- or CNN-based models, especially in irregular and sparse sensor networks; however, its reliance on recursive forecasting and heavy computational overhead poses limitations for real-time deployment.

Building upon the diffusion modeling paradigm, DiffSTG, proposed by Wen et al. [32], introduces further architectural innovations tailored specifically to the challenges of spatio-temporal graph forecasting. Unlike earlier approaches that separated spatial and temporal modeling stages, DiffSTG proposes a unified denoising network called UGnet that jointly captures multi-scale temporal dependencies and spatial correlations during the reverse diffusion process. UGnet combines a U-Net-like encoder-decoder structure—effective for hierarchical temporal feature extraction—with graph convolution layers to model local and global spatial relationships across the graph topology. This architectural design allows DiffSTG to iteratively refine noisy STG representations and generate highly accurate probabilistic forecasts, even under substantial uncertainty or missing data conditions.

Building on these foundations, Lin et al. [33] recently introduced SpecSTG, a novel framework that further expands the modeling capacity of diffusion-based approaches by operating in the spectral domain. SpecSTG leverages the graph Fourier transform to encode spatial dependencies into a frequency-based representation of the time series, enhancing the model's ability to capture both local and global graph structures. The architecture combines a modified GraphGRU encoder with a spectral graph WaveNet denoising module, employing an autoregressive diffusion process to predict the graph Fourier representation of future time steps. By operating in the spectral domain, SpecSTG offers a principled way to handle spatial complexity without excessively deep graph convolutions, improving stability and reducing noise in long-horizon forecasts.

## 2.4 The Rise of Foundation Models for Spatio-Temporal Forecasting

Recent advances in artificial intelligence have seen the emergence of foundation models [9]—large-scale, pre-trained architectures capable of adapting to a wide range of downstream tasks with minimal fine-tuning. Unlike task-specific architectures such as CNNs, LSTMs, or Spatio-Temporal Graph Neural Networks (STGNNs), which require bespoke design and training for each application, foundation models leverage vast amounts of data and general-purpose objectives to develop transferable representations. This shift is particularly compelling in the context of spatio-temporal forecasting, where traditional models often struggle with domain adaptation, limited supervision, and dynamic heterogeneity across space and time. By harnessing the generalization capabilities of foundation models, researchers aim to overcome these limitations and enable more scalable, flexible, and robust forecasting systems. Consequently, a new wave of work has emerged that explores how transformer-based architectures, pretrained language models, and multimodal encoders can be adapted to handle structured spatio-temporal data such as mobility traces, traffic flows, and environmental signals.

Zhou Tian et al. (2023) [24] proposed *One Fits All*, a unified framework for time series analysis that leverages pre-trained language models. Instead of training a new model from scratch, OFA freezes most of the pre-trained LM and only fine-tunes the embedding and normalization layers allowing the model to benefit from the vast amount of knowledge that has already been learned. This approach achieved state-of-the-art or comparable performance to existing methods in all tasks. *TrafficBERT [25]*, designed specifically for traffic flow forecasting, adopts a bidirectional transformer structure similar to BERT to predict overall traffic flow. This model is pre-trained with large-scale traffic data from various roads, increasing its generalization ability. The benchmarks in their paper demonstrate TrafficBERT's superior performance compared to traditional statistical models like ARIMA, LSTMs, GRUs, and Stacked AutoEncoders (SAEs)

LLMs can also be used complimentary along the STGs mentioned above to model interactions between different entities. A recent example would be TFM (Transportation Foundation Model) by Wang et al. (2023) [26], which takes a different approach by integrating the principles of traffic simulation into traffic prediction. It utilizes graph structures and dynamic graph generation algorithms to represent and model the interactions between various participants in a transportation system. As stated in their paper, "*The proposed approach shows promising results in accurately predicting traffic outcomes in an urban transportation setting.".*

## 2.5 Gaps and Challenges in Spatio-Temporal Forecasting

Despite the rapid evolution of forecasting models—from classical statistical approaches to advanced graph-based architectures—several fundamental limitations persist, hindering both performance and scalability in real-world spatio-temporal systems.

Traditional statistical models, such as ARIMA and SARIMA, have long served as the foundation for time series forecasting. However, these models inherently assume stationarity and linearity and are unable to capture spatial dependencies present in modern datasets. Their formulation treats observations as independent temporal sequences, disregarding interactions between locations or agents. This renders them ineffective in domains where spatial correlations play a critical role, such as transportation, climate, or mobility demand forecasting. Furthermore, they scale poorly when extended to multivariate or multi-location settings and offer little flexibility in adapting to evolving spatial dynamics.

Classical machine learning techniques, including support vector regression and gradient boosting machines, introduced greater modeling flexibility and nonlinearity. Nonetheless, these methods rely heavily on manual feature engineering to incorporate spatial and temporal cues. Since the learning process is agnostic to the underlying structure of spatio-temporal data, these models cannot natively capture interactions across space and time. Spatial features, when present, are often introduced as static inputs and fused with temporal features externally, making the models brittle in dynamic environments. Although their computational efficiency is generally favorable, their predictive capacity is constrained by the extent and quality of the hand-crafted features.

Early deep learning architectures, particularly LSTMs, CNNs, and temporal convolutional networks, advanced the state-of-the-art in modeling sequential dependencies. However, they too fall short in modeling spatial structure. Spatial context must be manually encoded—either through positional embeddings or auxiliary channels—and is not dynamically learned from data. Additionally, many of these models assume regular input structures, such as image grids or fixed-length sequences, which limits their applicability to irregular or graph-like spatial domains. The fusion of spatial and temporal information is often staged or static, preventing the model from jointly reasoning about spatio-temporal correlations in an adaptive manner.

Graph-based architectures, especially Spatio-Temporal Graph Neural Networks (STGNNs) and Spatio-Temporal Graph Convolutional Networks (STGCNs), offer a principled approach to learning from spatially structured data. These models construct graphs where nodes represent spatial entities and edges encode relationships such as proximity, flow, or correlation. Through this representation, STGNNs are capable of learning spatial dependencies alongside temporal patterns. However, this increased modeling capacity comes with substantial architectural complexity. Graph construction is often non-trivial, requiring careful design or external metadata. Moreover, the integration of graph convolutional layers with recurrent or temporal modules inflates model depth, leading to high memory requirements and long training cycles. As model complexity grows, interpretability and robustness often decline, and practical deployment becomes increasingly challenging.

Despite their notable strengths, diffusion-based spatio-temporal models still encounter key limitations. The probabilistic modeling of diffusion paths introduces substantial computational overhead during both training and inference, posing barriers to real-time deployment in latency-sensitive applications. Moreover, while spectral methods like SpecSTG offer compact and expressive representations, they may suffer when graphs are highly dynamic, as frequent changes in topology can destabilize spectral bases. These challenges highlight the growing need for more scalable, flexible, and generalizable approaches—paving the way for the exploration of foundation models for spatio-temporal forecasting.

Lastly, foundation models, including large language models and cross-modal transformers, have shown emergent capabilities in time series analysis and spatio-temporal prediction. However, these architectures were not explicitly designed for graph-structured data and generally lack spatial inductive biases. While transfer learning and prompt tuning offer promising avenues for adaptation, current implementations are constrained by their high resource demands and the need for extensive fine-tuning on domain-specific datasets. Furthermore, their interpretability, generalizability to unseen spatial contexts, and robustness to distributional shifts remain open research questions.

In summary, while each family of models addresses a particular facet of the spatio-temporal learning problem, none provide a holistic solution that balances spatial adaptivity, temporal fidelity, architectural simplicity, and computational efficiency. The trade-offs between model expressiveness and deployment feasibility remain a key challenge. Bridging this gap requires novel approaches that can integrate spatial and temporal signals efficiently, scale across heterogeneous domains, and operate within the practical constraints of real-time systems.

# 3 PROPOSED APPROACH

In this section, we define the shared mobility demand forecasting problem and the proposed approach. Our goal is to predict the demand of a city district $D$ at multiple future timestamps. That is our indicator of the shared mobility demand.

## 3.1 Problem Definition

Generally speaking, a spatio-temporal timeseries is a sequence of data points, where each one is associated with a time index and a spatial location. For the purposes of our work, an area of interest (a city, a region, etc.) is partitioned into a set of geographical districts, where each district $A$ is represented as a polygon.

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ denote a graph where nodes $\mathcal{V}$ represent the districts and edges $\mathcal{E}$ encode the physical proximity between the districts (neighbor, non-neighbor). We denote $X \in \mathbb{R}^{T\times N\times 3}$ as a multivariate timeseries of mobility demand observations collected at districts $A \in \mathcal{V}$ over $T$ timesteps, where each feature vector is of size 3 (timestamp, district, demand). As such, we denote $D_t^A$ as the demand of micro-mobility vehicles within district $A$ at time $t$. Given $\tau \in T$ past observations for district $A$, we construct its spatio-temporal timeseries as $X_t^A = \{D_{t-\tau}^A, D_{t-\tau+1}^A, \dots, D_{t-1}^A, D_t^A\}$. The problem addressed in this thesis is to predict the micro-mobility demand value $D_{t+m}^A$ for district $A$, $m$ time steps ahead, given $X_t^A$ and graph $\mathcal{G}$, i.e. learn the function $f$ such that

$$\widehat{\mathrm{X}}_{\mathrm{T+1:T+H}} = \mathrm{F}(\mathrm{X}_{1:\mathrm{T}}, \mathrm{G}),$$

where $\widehat{\mathrm{X}}_{\mathrm{T+1:T+H}}$ are the predicted demand values over a forecast horizon $H$.

To ground the above definition in a real-world scenario, we illustrate an example derived from our dataset (detailed in Section 4). As shown in Figure 3-1, the forecasting task involves predicting the future values (orange) of each district's mobility demand time series, given its current and historical observations (blue), while incorporating relevant temporal, spatial, and contextual features. Accurate forecasting requires uncovering latent patterns—such as seasonality and recurring trends—embedded within these time series. For instance, the demand profiles of the two major districts in Rotterdam reveal consistent commuting patterns, with the highest and most prolonged peak occurring on Fridays, likely due to the convergence of work and leisure activities.

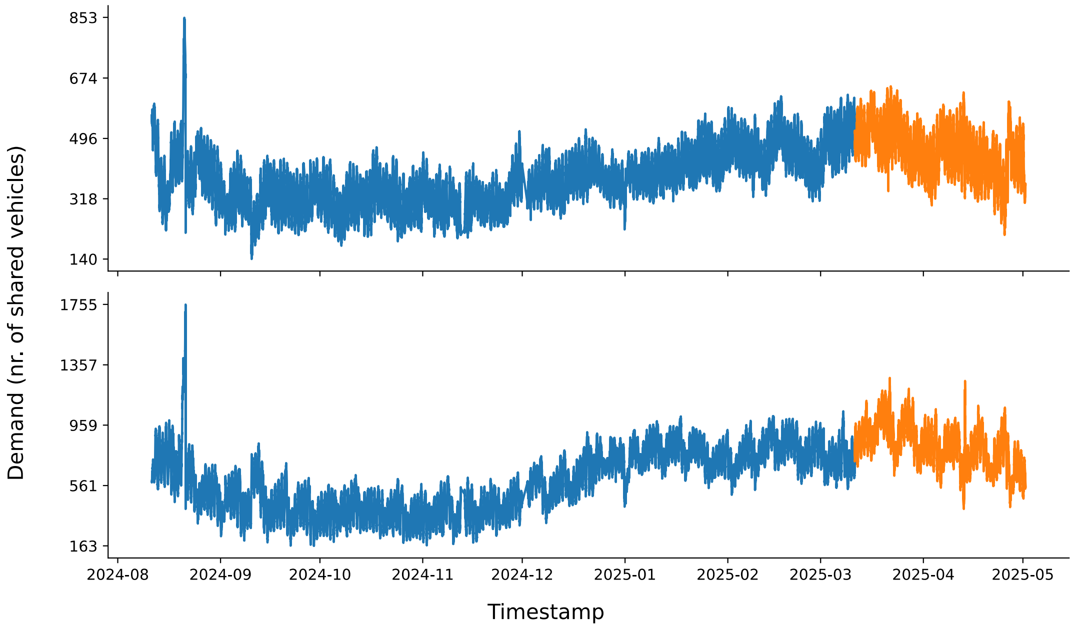


**Figure 3-1: Two indicative timeseries for two regions of Rotterdam: Delfshaven (top) and Rotterdam Centrum (bottom). The x-axis denotes time, while the y-axis indicates the measured demand values.**

These patterns are further highlighted in Figures 3-2 and, which visualize the interaction between *hour-of-day* and *day-of-week* features and the seasonal demand evolution in the Rotterdam Centrum district. One can easily distinguish the commute patterns in each workday (concentrated peaks around midday) and the leisure use on the weekends with a more spread-out peak demand around the middle of the days.

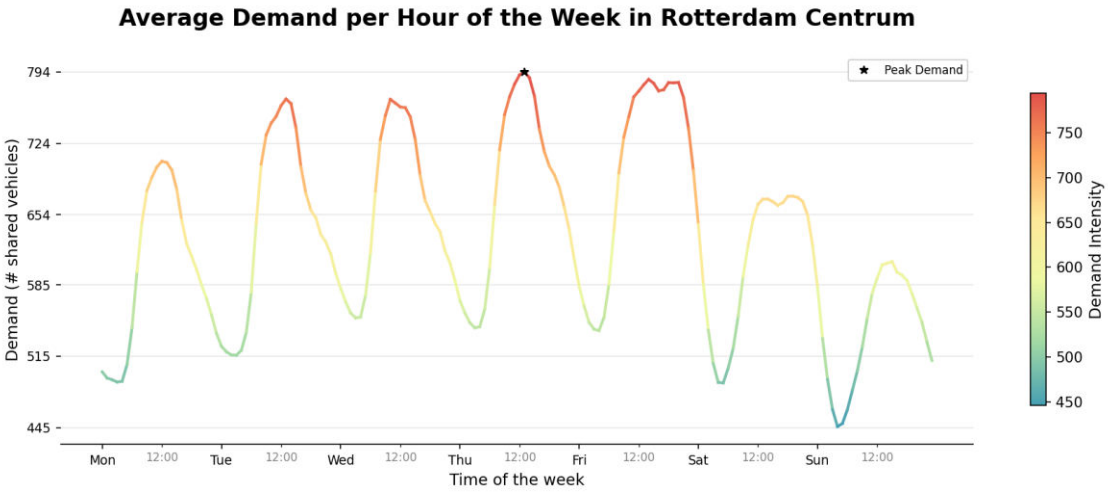


**Figure 3-2: Average hourly demand across the week in Rotterdam Centrum. The coloring indicates the relative demand intensity and, along with the peaks, aid in highlighting daily and weekly usage patterns.**

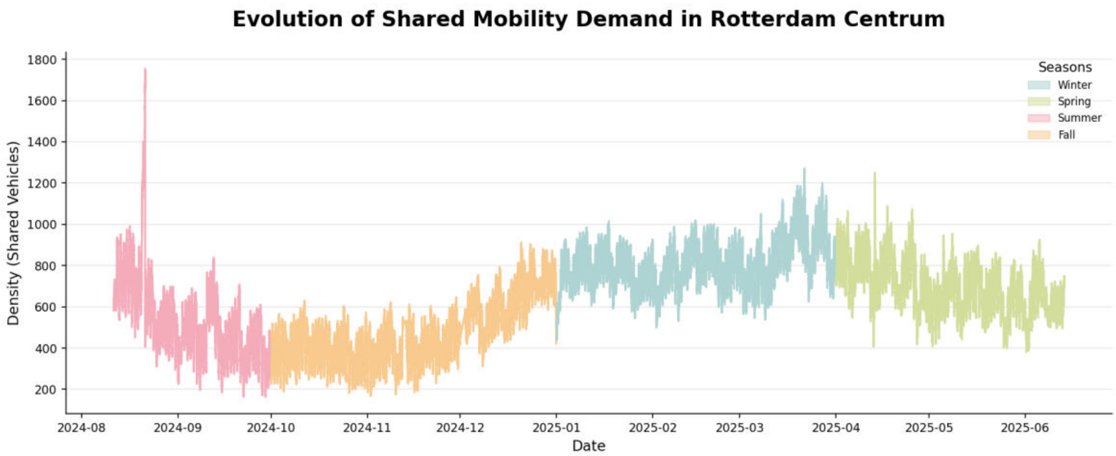


**Figure 3-3: Temporal evolution of the shared mobility demand in Rotterdam Centrum. Seasons are color-coded to highlight the different behaviors. In Fall and Winter, demand patterns are more stable whereas during Spring and Summer short-term spikes are observed often at "random" periods.**

## 3.2 Methodology

We propose the Gradient Boosting Demand Predictor (*GBDP*) framework, designed to accommodate both classification and regression objectives depending on the nature of the downstream task.

Gradient Boosting is a powerful ensemble learning technique introduced by Jerome Friedman in the late 1990s, initially formulated as Gradient Boosting Machines (GBM). It builds predictive models in a stage-wise fashion by iteratively adding decision trees—typically shallow ones—as weak learners. Each tree is trained to minimize the residual errors made by the ensemble of all previous trees, using a gradient descent-like procedure in function space. The core idea is to fit new models to the negative gradients of a specified loss function (e.g., squared error for regression or cross-entropy for classification), thereby incrementally improving the overall prediction. This approach makes gradient boosting particularly effective at capturing complex, non-linear relationships in the data.

Gradient Boosting has been selected as the backbone algorithm for this study owing to its strong empirical performance and several advantageous characteristics [10] that align well with the nature of our problem. First, it excels at capturing complex non-linear relationships, which are prevalent in urban mobility patterns driven by human behavior, infrastructure constraints, and environmental conditions. Its robustness to noise and partial observations is another critical advantage, especially in real-world data acquisition scenarios where GPS inaccuracies or data sparsity may occur.

One of the most compelling reasons for employing Gradient Boosting is its demonstrated effectiveness on structured, tabular data. Our feature set—comprising spatio-temporal signals and engineered time-series features such as lags, rolling statistics, and exponentially weighted trends—is well-aligned with this strength. Additionally, Gradient Boosting offers intrinsic model interpretability through feature importance metrics, enabling deeper insights into the predictive contribution of temporal and spatial indicators. Moreover, the algorithm is known for its computational efficiency.

Modern implementations as XGBoost[1] (the one used in our approach), LightGBM[2], and CatBoost[3] have further refined the method by introducing regularization, tree-pruning, parallelization, and advanced handling of categorical and missing values—rendering it one of the top-performing algorithms for structured data across a wide range of machine learning tasks. They are optimized for speed and scalability, allowing the training of multiple models within reasonable computational budgets. This is particularly beneficial in time-sensitive applications where rapid model iteration and deployment are required. Finally, compared to deep learning models, Gradient Boosting offers a more accessible hyperparameter space and faster training times, facilitating quicker experimentation without the extensive tuning often required by neural networks.

In our case, the model is implemented in two primary variants:

- Classification-Based GBDP: Each model is trained to classify future demand into discrete ordinal categories—e.g., *Low, Medium, or High*—based on demand thresholds computed from historical distribution statistics.
- Regression-Based GBDP: This variant aims to predict the exact numerical value of future demand, formulated as a standard supervised regression task.

Given an urban area partitioned into $M$ spatial units (districts), and a multi-step forecasting objective spanning $T$ future horizons (5min, 15min, 30min, 60min), the proposed architecture centers around a single global model, trained jointly across all districts of a given city. This unified approach, termed the Global Gradient Boosting Demand Predictor (Global-GBDP), is designed to leverage shared spatio-temporal regularities across regions while retaining district-specific nuances via the inclusion of the district identifier as an input feature. The ID is encoded and integrated into the feature space, allowing the model to implicitly learn distinct behavioral patterns per district. As illustrated in Figure 3-3: Global-GBDP architecture, the first step of our methodology involves partitioning the area of interest into a set of districts, which are then represented as nodes in an adjacency graph. Two districts are considered neighbors if they share a common boundary[4]. This graph-based representation allows us to formally capture local spatial dependencies among districts.

The global model adopts a broader perspective. Instead of focusing solely on a district and its neighbors, it aggregates the time series from all districts within the selected metropolitan area. These are organized into a single multi-indexed dataset, where the first index is associated with a corresponding district identifier and the second level of indices corresponds to the timestamps.

[1] https://xgboost.readthedocs.io
[2] https://lightgbm.readthedocs.io
[3] https://catboost.ai
[4] The inclusion of neighboring districts is justified by the limited spatial range and temporal duration of trips usually made by e-bikes and e-scooters. Due to their physical constraints, micro-mobility vehicles are unlikely to reach distant districts within short timeframes, making the consideration of neighboring districts only being a sufficient choice for effective spatio-temporal modeling.

This design enables the model to exploit patterns across regions and supports a single model trained jointly on the entire spatial domain.

## 3.3 Feature Extraction

To effectively model the spatio-temporal patterns present in shared micro-mobility demand, careful feature engineering is essential. Our approach builds upon the formal problem definition (Section 3.1) and the model architecture (Section 3.2), transforming raw time series data into a rich set of predictive variables. The features are designed to encapsulate temporal dynamics (e.g., daily and weekly seasonality), spatial influence (from neighboring districts), and contextual factors (such as holidays or weather conditions, if available). Regardless of whether the model is specialized (district-specific) or generalized (area-wide), we construct the same set of feature matrices that include lagged observations, spatial context, and time-based encodings. These features serve as the input to both the classification and regression branches of our Gradient Boosting Demand Predictor. The feature pipeline can be seen in Figure 3-3. Given a timeseries $X_t^A$ of a target district $A$, we perform feature extraction to enhance the spatio-temporal modeling capacity of our method.

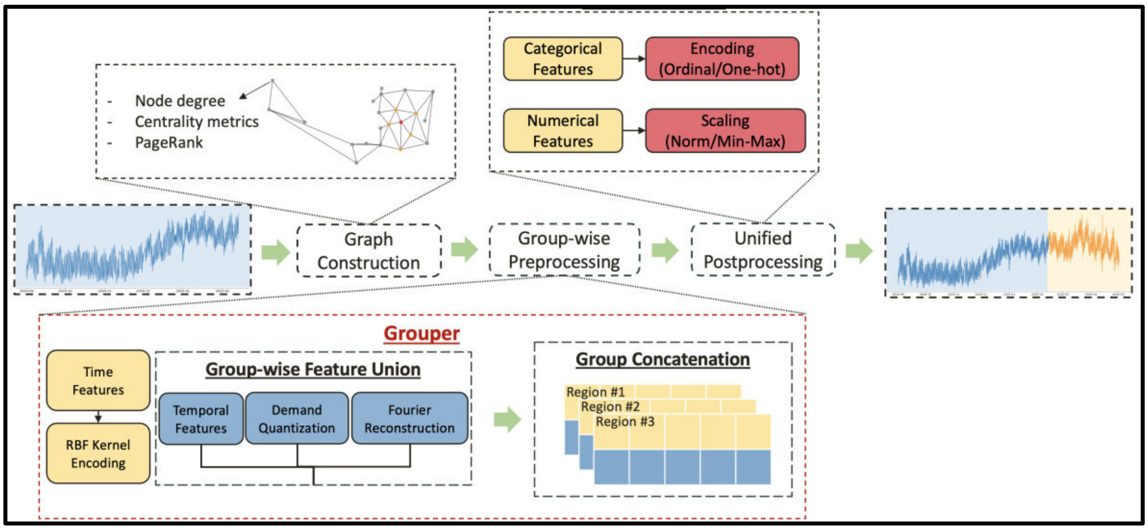


**Figure 3-4: Global-GBDP architecture**

All extracted features used in our methodology are systematically summarized in Table 3-1, which provides an overview of their definitions, temporal scope, and intended purpose within the prediction framework.

### 3.3.1 Temporal Features

Temporal features are organized into four primary categories: time-related, lagged, rolling, and exponentially weighted features. These features are derived not only from the target district but also from its neighboring districts, thereby incorporating both spatial and temporal dependencies.

Time-related features are designed to uncover and exploit recurring temporal structures in the demand time series, which are often governed by human mobility behavior and urban activity cycles. These features are vital for capturing periodic patterns that influence the target variable at multiple temporal resolutions. Hour of the day (ranging from 0 to 23) reflects the daily cycle of human activity. This feature is crucial for identifying well-established diurnal patterns such as morning and evening commute peaks and late-night drops in mobility. Day of the week (ranging from 0 to 6, where 0 denotes Monday) captures variations in demand associated with the structure of the week. Weekdays typically exhibit different mobility behaviors compared to weekends—e.g., higher morning activity during weekdays due to work-related trips, and more erratic or leisure-related patterns on weekends. Month of the year (0 to 11, where 0 corresponds to January) accounts for seasonal effects in mobility, such as increased bike-sharing usage in summer months or lower activity during winter. These monthly variations can be particularly important in regions with strong weather seasonality or tourist influxes that vary by season. Minute of the hour (0 to 59) provides finer temporal granularity, capturing intra-hour dynamics such as bursts of demand during the beginning or end of scheduled events (e.g., classes, work shifts). This component is especially relevant when using high-frequency data as it allows the model to learn short-term temporal regularities that may be obscured at coarser levels.

**Table 3-1: Summary table of the extracted features used**

| Category | Feature Name | Description |
|---|---|---|
| Base | district | Geographic or administrative area identifier |
| | crowd | Raw crowd measurement |
| Calendar | hour_rbf_* | RBF encoding of the hour of the day (0-23) |
| | weekday_rbf_* | RBF encoding of the day of the week (0-6) |
| | month_rbf_* | RBF encoding of the month (1-12) |
| | quarter_rbf_* | RBF encoding of the quarter (1-4) |
| Lagged | crowd_lag_{1, 5, 10, 15, 1440} | Crowd values from previous time steps |
| Rolling | crowd_rolling_mean_{5,10, 15} | Rolling mean over past n time steps |
| EWMA | crowd_ewm_{5, 10, 15} | Exponentially weighted moving average of the target variable |
| Statistical | crowd_diff | First-order difference (current - previous) |
| | crowd_diff_rate | Relative change to previous value |
| | crowd_pct_change | Percentage change compared to previous time step. |
| | crowd_deviation | Deviation from the average demand |
| | crowd_change | Label-based change between consecutive demands |
| | crowd_adjusted | Scaled feature that considers the relevant change label |
| Fourier | fourier_crowd | Fourier-transformed component to capture periodic patterns in crowd data. |
| Graph | node_degree | # of connections of a node in the graph. |
| | degree_centrality | Normalized degree |
| | closeness_centrality | Measures how close a node is to all others |
| | betweenness_centrality | Measures how often a node lies on shortest paths between other nodes. |
| | eigenvector_centrality | Importance of a node based on its neighbors' importance. |
| | pagerank | Google's PageRank algorithm score |
| | clustering_coef | Measure of how connected a node's neighbors are (triangle-based). |

The second category of features, referred to as lagged features, is designed to model the temporal dependencies between past observations of the time series and its current state. These features play a critical role in enabling the model to learn from historical demand fluctuations and their influence on future values. For each district, a set of lagged features is constructed by systematically shifting the original time series backward along the time axis. Formally, given a time series $X_t^A$ representing the observed demand in district $A$ at time $t$, a lagged feature at temporal offset $\tau$ is defined as $X_{t-\tau}^A$. This process effectively encodes past system states as additional inputs for the learning algorithm. To comprehensively capture both short-term and medium-term dependencies, lagged features are computed at multiple offsets—specifically at 1, 5, 10, and 15 minutes prior to the current timestamp. This multi-resolution design allows the model to leverage high-frequency variations (e.g., sudden demand surges) as well as more sustained trends (e.g., gradual build-up before rush hours). Incorporating lagged features into the feature set provides the predictive model with a memory of recent system dynamics. This enhances the model's ability to detect recurrent patterns, local trends, and autocorrelations that are inherently present in spatio-temporal mobility data, ultimately improving the robustness and accuracy of future demand forecasts.

The third category, rolling features, captures short-term trends and localized variability in the demand time series by applying statistical aggregation over recent temporal windows. These features provide a smoothed representation of recent activity, helping the model to mitigate noise while highlighting emerging patterns that may influence near-future predictions. Rolling features are computed using a sliding window mechanism, whereby a window of fixed length moves incrementally over the time series. Within each window, summary statistics are extracted to represent the behavior of the system over the immediate past. In this study, we apply rolling windows of varying sizes—specifically 5, 10, and 15 minutes—to account for different levels of temporal granularity. The rolling mean is the primary statistic used—though other statistics can be used alongside (e.g., standard deviation, variance, kurtosis)—, offering insight into the local central tendency of demand. Smaller windows emphasize fine-grained fluctuations (e.g., demand surges in response to an event), while larger windows smooth over these variations to reveal broader patterns (e.g., the general build-up toward peak hours). Optionally, other measures such as rolling standard deviation or rolling minimum/maximum can be incorporated to further characterize local volatility and extremes, although the current implementation prioritizes interpretability and computational efficiency. By incorporating rolling features, the model gains access to a dynamic summary of recent temporal context, which improves its sensitivity to evolving demand patterns while reducing the impact of short-term anomalies or irregularities.

Exponentially weighted features aim to capture temporal dependencies while adaptively emphasizing more recent observations. Unlike simple rolling statistics that treat all values within a window equally, these features assign decaying importance to older data points, thereby ensuring the model remains responsive to recent changes while still incorporating longer-term patterns. A key technique employed in this category is the Exponentially Weighted Moving Average (EWMA). Formally, for a given time series $X_t$, the EWMA at time $t$ is computed recursively as $\mathrm{EWMA}_t = \alpha \mathrm{X_t} + (1-\alpha)\mathrm{EWMA_{t-1}}$ where $\alpha \in (0,1]$ is the smoothing factor that controls the rate of exponential decay. The value of $\alpha$ is typically parameterized as $\alpha = \frac{N+1}{2}$, where $N$ denotes the effective window size. Higher values of $\alpha$ result in greater sensitivity to recent fluctuations, while lower values ensure smoother, more stable trends. This approach enables the model to quickly adapt to sudden changes in mobility demand, which may be induced by unpredictable external factors such as road closures, public events, or inclement weather, without entirely discarding the influence of historical context. Compared to fixed size rolling windows, exponentially weighted statistics offer a more nuanced trade-off between adaptivity and stability.

To accommodate the wide range of demand magnitudes observed across districts—from the order of $10^1$ to $10^3$—we introduce an **adjusted demand transformation**. This transformation normalizes the relative scale of demand while preserving its semantic interpretation, allowing the model to more effectively handle disparities across regions and time. The process begins by quantizing the raw demand values into $N$ ordered categorical levels, typically centered around a median label (e.g., $Low < Medium < High$). Each level is then associated with a scaling multiplier derived from a predefined scheme governed by a scaling factor $s$, which determines the magnitude gap between adjacent levels. The main idea for the scaling rule is to assign the central level (e.g., $Medium$) a multiplier of 1 and successively decreasing (increasing) multipliers for lower (higher) levels by dividing (multiplying) by $s$ (e.g., $Low \rightarrow 1/s$ , $High \rightarrow s$). The adjusted demand $\tilde{D_t}$ is then computed by multiplying the original raw demand $D_t$ by the corresponding multiplier $\gamma$ associated with its quantized level: $\tilde{D_t} = \gamma(D_t) \cdot D_t$

To further enhance the model's sensitivity to short-term dynamics, a suite of differential features is computed, capturing both the absolute and relative nature of temporal changes in demand. These features are derived from the raw and adjusted demand signals and aim to highlight abrupt fluctuations, emerging trends, and local variability in the data. Specifically, we include the first-order difference, which reflects the absolute change in demand between consecutive observations and helps detect sudden increases or drops. To capture the acceleration or deceleration of demand changes, we also compute the second-order difference, offering insight into how the pace of fluctuation evolves over time. In addition to absolute changes, we introduce the percentage change feature, which expresses the relative variation between consecutive time points and provides a scale-independent view of demand volatility. Furthermore, we calculate the deviation of each observation from its corresponding 5-minute rolling mean, which serves to isolate transient anomalies or shifts from the underlying short-term trend. These differential signals, when combined with other temporal and spatial features, enrich the input representation and improve the model's ability to recognize fine-grained patterns in demand evolution.

**Algorithm 1: Demand Scaling Multipliers**

```
Input:
N. ← Total number of demand levels (odd integer)
s  ← Scaling factor (e.g., 1.5 or 2.0)

Output:
multipliers ← Dictionary mapping each demand level to a multiplier

Procedure:

Let mid ← floor(N / 2)          // Index of the central (neutral) level
multipliers ← empty dictionary
for i from 0 to N-1 do
      if i == mid then
            multipliers [i] ← 1.0
      else if i < mid then
            multipliers [i] ← 1 / s^(mid − i)
      else
            multipliers [i] ← s^(i − mid)
      end if
end for
return multipliers
```

### 3.3.2 Spatial Features

To enhance the model's spatial awareness and capture inter-district dependencies, we incorporate graph-based features derived from the urban topology of the area of interest. Conceptually, the forecasting task can be framed as a graph learning problem, where each district is represented as a node, and the goal is to predict the future demand at each node based on both its own historical behavior and its structural position within the urban graph.

To construct this graph, we begin by transforming each district polygon into a representative point using its geometric centroid. Edges between nodes are then established based on spatial proximity or shared boundaries, effectively modeling neighborhood relationships. Figure 3-4 illustrates this transformation from geographic regions to a graph structure.

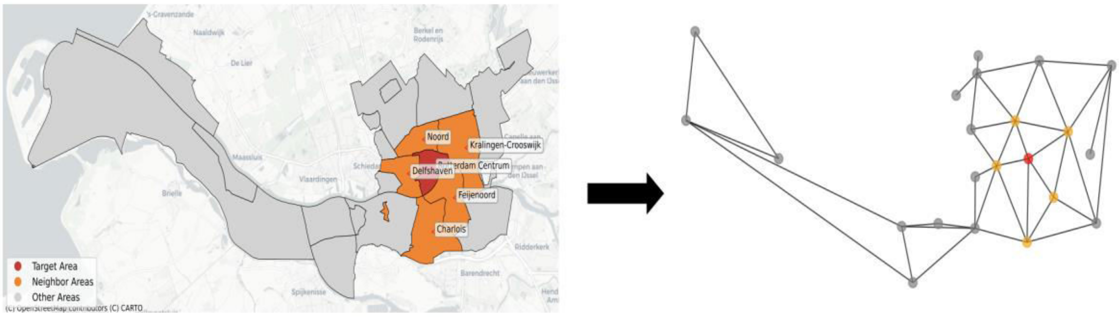


**Figure 3-5: Polygon-to-graph construction. The graph maintains connections between a target district-node (red) and its neighboring districts-nodes (orange)**

Once the graph is constructed, we compute a suite of node-level graph features that provide rich structural context for each district. These include:

- **Degree**: the number of immediate neighboring districts, reflecting direct spatial connectivity.
- **PageRank**: a measure of node influence based on recursive importance propagation across the graph.
- **Closeness centrality**: quantifies how quickly information (or demand signals) can spread from one district to all others.
- **Betweenness centrality** captures the extent to which a district lies on the shortest paths between other district pairs, indicating its role as a transit or connector hub.
- **Eigenvector centrality** measures the influence of a district based not only on its direct connections but also on the importance of its neighbors.
- **Clustering coefficient**: reflects the degree to which a district's neighbors are interconnected, indicating local cohesion.

By integrating these graph-derived features into our predictive framework, we enable the model to incorporate spatial topology and cross-district interactions, which are often critical drivers of urban mobility dynamics.

### 3.3.3 Statistical Features

**Statistical features** serve to enrich the temporal representation of the target variable by capturing nuanced patterns of change, volatility, and deviation over time. These features quantify both the direction and magnitude of fluctuations across observations, offering valuable insight into underlying temporal dynamics. Examples include differencing-based metrics, such as the difference between subsequent demands and the rate with which it is happening, which highlight shifts between consecutive values and help reduce trend-induced bias—facilitating the modeling of non-stationary series.

They can also capture general deviations from expected baselines—often relative to moving averages or seasonal expectations. Such features are particularly useful for detecting anomalies, contextualizing current states, and modeling behavioral regimes that reflect real-world disruptions or cyclical patterns. Collectively, these transformation-based statistical features enhance the model's sensitivity to **short-term dynamics and contextual variations**, strengthening its ability to generalize across complex urban mobility patterns and fluctuating demand regimes.

### 3.3.4 Fourier-Based

**Fourier-based** features aim to distill the essential cyclical structure of a time series by removing short-term noise and isolating dominant periodic components. These features are derived through the application of the **Fourier Transform**, which decomposes the original signal into a weighted sum of sine and cosine functions across various frequencies. By reconstructing the time series using only the most significant low-frequency components, the resulting approximation captures long-range, smooth temporal patterns—such as daily or weekly seasonality—while attenuating transient fluctuations.

This form of transformation is particularly beneficial in mobility demand modeling, where recurrent commuting patterns, event-driven surges, or diurnal cycles are often embedded within a noisy signal. The Fourier-based representation thus provides a compressed yet informative view of temporal behavior, offering the model access to latent periodic signals that may not be apparent through raw observations alone. Integrating these features into the learning pipeline enables the model to leverage stable, frequency-based abstractions of demand evolution, complementing lagged and statistical indicators with a more global, harmonically-informed perspective.

## 3.4 Evaluation Metrics

To assess the predictive performance of the proposed Gradient Boosting Demand Predictor (GBDP) across both regression and classification tasks, we employ a suite of well-established evaluation metrics, each chosen to capture different aspects of model behavior.

For the regression variant of GBDP, where the objective is to forecast the actual demand values for each district-horizon pair, we use the following metrics:

- **Mean Absolute Error (MAE):** MAE measures the average magnitude of prediction errors without considering their direction. It is particularly useful for applications where absolute deviations are of direct practical relevance, such as mobility resource planning.
- **Root Mean Squared Error (RMSE):** RMSE penalizes larger errors more than MAE due to the squaring operation, thus providing insight into the variance of prediction errors. It is especially appropriate for capturing the effect of outliers or high-variance events in urban mobility patterns. It is defined as:
- **Symmetric Mean Absolute Percentage Error (sMAPE):** The SMAPE metric is designed to account for the relative difference between predicted and actual values, considering their magnitudes in a balanced manner. Unlike other error measures, it assigns equal weight to overestimations and underestimations. The formula for calculating SMAPE is defined as follows:
- **Coefficient of Determination ($R^2$ Score):** Measures the proportion of variance in the true values that is predictable from the inputs. It offers a normalized view of model fit, with values closer to 1 indicating stronger explanatory power.

For the classification variant, which maps future demand levels into discrete categories (e.g., Low, Medium, High), we evaluate performance using:

- **Accuracy**: The proportion of correctly classified instances,
- **Precision**: The correctness of positive predictions per class,
- **Recall**: The model's ability to identify all relevant instances
- **F1-Score**: Harmonic mean of precision and recall, capturing their balance

All classification metrics are computed in their macro-averaged form to ensure equal weight across all demand levels, regardless of class imbalance. This approach provides a holistic view of the model's classification capacity across different demand regimes.

# 4 EXPERIMENTAL STUDY

This section describes the data and preprocessing steps used in our experimental study (Section 4.1) as well as our findings (Section 4.3). The experiments were conducted on a MacBook M3 Air with 16GB of RAM, 8 CPU cores and 10 GPU Cores

## 4.1 Experimental Setup

### 4.1.1 Data Sources

In this study, we utilize real-time shared micro-mobility data provided by *Deelfiets Nederland*, a prominent micro-mobility service operator in the Netherlands. The data is accessed through an open API[5], which exposes the live GPS locations of publicly available vehicles across the country. Each API response includes structured geolocation records for individual vehicles, with metadata such as GPS coordinates, vehicle type (e.g., bicycle, moped, cargo) and provider. These records will be aggregated to derive region-level demand counts depending on temporal and spatial granularity requirements.

To ensure consistent temporal sampling, our data acquisition pipeline queries the API every 60 seconds, creating a dense time series suitable for modeling fine-grained demand fluctuations. The collection process is fully automated and handles scheduled execution of API requests, data parsing, and downstream storage operations. Responses are manually timestamped with the one that each request was made – since temporal information is absent - and stored in a structured database, ensuring reproducibility and scalability for downstream processing. The above result in a high-frequency spatio-temporal dataset. Our focus is on three major metropolitan areas—Amsterdam, Rotterdam and The Hague—each of which is partitioned into multiple administrative districts based on city-defined region boundaries (see Figure 4-1). The temporal span of the collected data is approximately 9 months for Rotterdam (August 11, 2024 – May 1, 2025), 6 months for Amsterdam (November 11, 2024 – May 1, 2025) and 5 months for The Hague (December 11, 2024 - May 1, 2025) providing adequate historical context for both short- and medium-term spatio-temporal forecasting.

Once collected, the raw GPS-level data undergoes a multi-stage preprocessing pipeline designed to transform noisy, real-time snapshots into structured spatio-temporal demand time series. The primary transformation step involves filtering and removing the traces that fall outside our desired areas of interest (AoI).

To derive demand estimates at the district level, a spatial filtering and aggregation procedure is applied. First, a point-polygon intersection is performed to associate each mobility trace with its corresponding district boundaries, discarding traces that fall outside the predefined study areas. Next, the filtered traces are aggregated based on their unique spatio-temporal identifiers (i.e., timestamp and district ID), summing the occurrences within each region to obtain demand estimates. This transformation preserves the spatial and temporal integrity of the data while ensuring compatibility with the forecasting models. In its final form, the dataset consists of 22,607,077observations across 97 districts in Amsterdam, 5,790,422 observations across 22 districts in Rotterdam and 8,310,254 from The Hague.

[5] https://api.deelfietsdashboard.nl/dashboard-api/public/vehicles_in_public_space

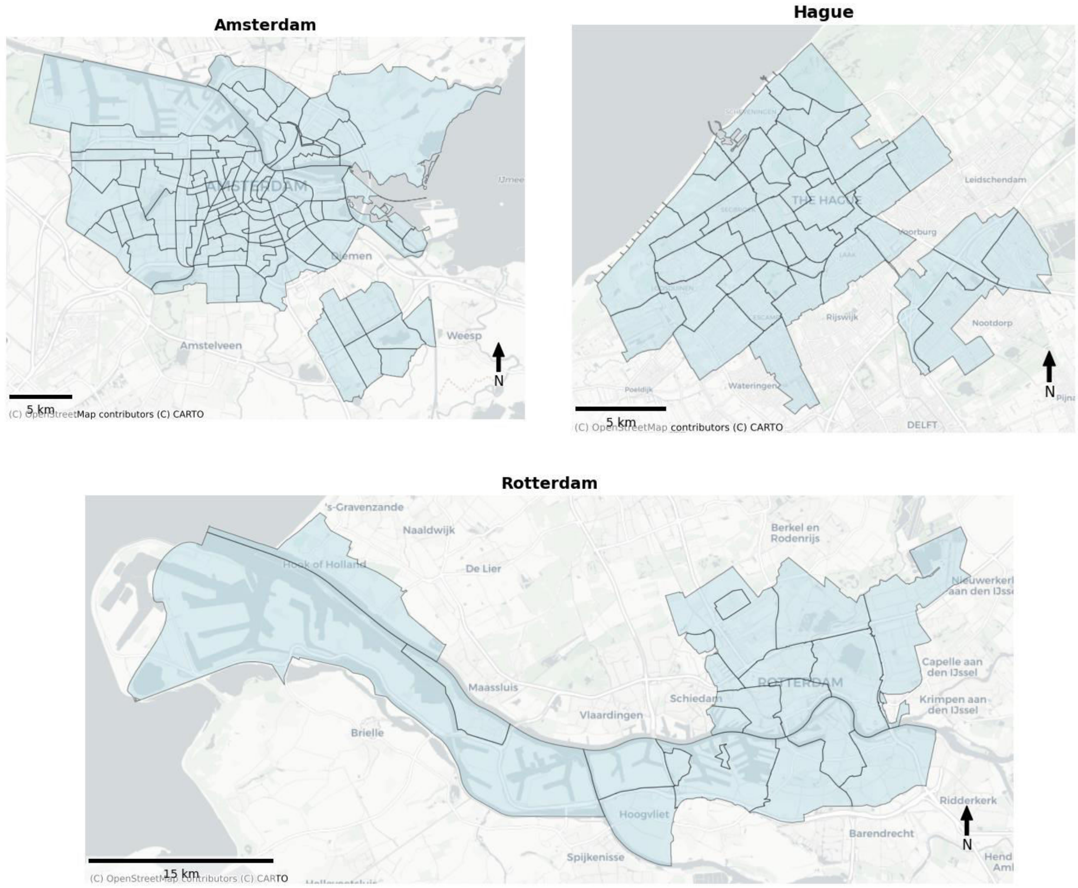


**Figure 4-1: Administrative regions of Amsterdam (left), Hague (right) and Rotterdam (bottom)**

### 4.1.2 Preprocessing

To ensure the robustness and reliability of the predictive framework, we employ a variance-based filtering mechanism to exclude spatial units with minimal activity variability. Specifically, districts exhibiting low temporal demand variance (threshold < 20) are excluded from the dataset. This step is motivated by the need to eliminate areas with either sparse or nearly constant micro-mobility activity, which tend to introduce noise, contribute to data imbalance, and reduce the generalizability of learned patterns.

However, raw variance may be artificially inflated due to sporadic outliers or anomalous spikes in demand—such as those induced by data errors or rare events (e.g., festivals, protests, outages). To mitigate this risk, we apply an Interquartile Range (IQR)-based outlier filtering procedure to each district's demand time series before computing its variance. This preprocessing step ensures that only genuinely low-variability districts are removed, while districts with legitimate but irregular fluctuations are retained for analysis. This variance-informed filtering strategy helps reduce model overfitting to uninformative regions and improves training efficiency by focusing learning capacity on districts that exhibit dynamic and meaningful demand patterns.

We now obtain a refined subset of districts from both Amsterdam and Rotterdam, each exhibiting sufficient spatio-temporal demand variability to support robust forecasting. In total, $M^{\mathrm{A}} = 88$ districts from Amsterdam, $M^{\mathrm{R}} = 13$ districts from Rotterdam and $M^{H} = 43$ from The Hague satisfy the variance criterion after outlier filtering. These districts represent the most active and dynamically evolving areas of each city's shared mobility landscape, characterized by distinct temporal rhythms (e.g., rush hour peaks, weekend troughs) and spatial heterogeneity (e.g., commercial centers vs. residential zones).

To validate our models, we employ a temporal-based split of the time-indexed dataset using a 60-20-20 ratio for the train, validation, and test sets, respectively. This method preserves the temporal integrity of the data, preventing leakage of future information and ensuring a realistic evaluation of model performance. In preparation for training the classifier variant of our architecture, an additional preprocessing step was applied to transform the continuous demand values into discrete categories. Specifically, we employed a quantile-based discretization function to define three demand classes: Low, Medium, and High. This transformation improves class separability and mitigates issues associated with class imbalance, which is common in naturally skewed demand distributions.

The definition of the classes is based on the relative deviation from the maximum observed daily demand across the dataset. Let the "peak" daily demand be scaled to 100 and let $d \in (0, 50)$ represent a symmetric tolerance threshold. We define m*edium* demand the values within the interval $[50 - d, 50 + d)$ and *low* (*high*) demand the values below (above) $50 - d$. In our experiments, we set $d = 20$, resulting in three equally sized quantile-based intervals. This threshold provides a balanced distribution of samples across demand levels, which facilitates robust classification.

## 4.2 Training Setup

### 4.2.1 Training Configuration

Our forecasting task is formulated as a multi-step ahead prediction problem, targeting future demand values at four distinct horizons: 5, 15, 30, and 60 minutes into the future. For each district, we generate four parallel target variables corresponding to each horizon, allowing the model to learn both short- and long-term temporal dynamics.

To construct the training labels, the original time-indexed demand series is shifted forward by each horizon independently. For a given time index $t$ and forecast step $h$, the model is trained to predict the value at $t + h$ using features available at or before $t$. This results in four distinct supervised learning tasks per district in the specialized models, or four multi-target regression/classification heads in the global architecture. Unlike recursive strategies that propagate prediction error through time, we adopt a direct forecasting approach, where each horizon is predicted separately. This ensures the stability and reliability of each forecast without accumulating horizon-wise errors.

### 4.2.2 Hyperparameter Optimization

To fine-tune the performance of the Gradient Boosting Demand Predictor (GBDP) models, we adopt a Bayesian Optimization strategy implemented through the Optuna framework. Bayesian Optimization is particularly well-suited for hyperparameter tuning in scenarios involving computationally expensive model training, where exhaustive grid or random search would be inefficient and resource-intensive. This probabilistic approach constructs a surrogate model of the objective function to model the relationship between hyperparameters and the evaluation metric. It iteratively selects the next set of parameters to evaluate by maximizing an acquisition function, such as Expected Improvement (EI), which intelligently balances exploration (searching new regions of the hyperparameter space) and exploitation (refining promising areas based on previous trials).

By incorporating prior trial outcomes, Bayesian Optimization reduces the number of iterations required to converge toward near-optimal configurations. This allows for a resource-efficient search, which is crucial given the large number of models trained (one per district and prediction horizon, or a global one for all). The automatic adjustment of the exploration–exploitation trade-off via the acquisition function also enhances its adaptability to varying search spaces and task-specific performance landscapes. We specifically employ two key components from Optuna to enhance the tuning process: *TPESampler* and *MedianPruner*.

The **TPESampler** (Tree-structured Parzen Estimator) replaces traditional Gaussian Processes with a non-parametric density estimation approach that models good (and bad) hyperparameter configurations separately. By maintaining two distributions—one for promising and one for unpromising trials—it estimates the likelihood of improvement and guides the sampling process toward regions of the space with high expected performance. This allows the optimization algorithm to adaptively balance exploration (testing new areas) and exploitation (refining known good regions), making it particularly well-suited for high-dimensional, irregular search spaces.

The **MedianPruner** introduces an early stopping mechanism that terminates underperforming trials based on intermediate validation results. Specifically, if a trial's performance at a given step falls below the median of previous completed trials, it is pruned. This reduces unnecessary computation on suboptimal configurations and accelerates convergence toward better solutions.

To further refine the hyperparameter selection process, we implement a two-phase optimization strategy. In the first phase, we define a broad hyperparameter space with wider value intervals. This allows the TPESampler to explore a large, diverse region of the search space and identify a promising sub-region where good-performing configurations tend to cluster. This phase helps prevent early convergence to local optima. Based on the best-performing trial from Phase 1, we redefine the search space by using its hyperparameters as center points. We then perform a second round of optimization within a narrower neighborhood, defined by smaller incremental steps around those center values. This increases the resolution of the search and helps capture near-optimal configurations that may have been missed in Phase 1 due to coarse granularity.

This coarse-to-fine search paradigm ensures a balanced approach—efficiently exploring the space without skipping over subtle but high-performing regions. Each phase includes a defined number of trials ($[100, 500]$ recommended for Phase $1, [30, 100]$ for Phase 2 depending on optimization importance). Quality visualization metrics – as shown in Figure 4-2 – are displayed at predefined intervals to monitor the process of tuning, examining it from different aspects and providing insights / estimations on unexplored sub-spaces to assist in the case of manual tuning. Some of them include optimization history, top $N$ parallel coordinate for each hyperparameter, contour plot of the top 2 hyperparameters etc. Fig. represents a snapshot of the above monitoring

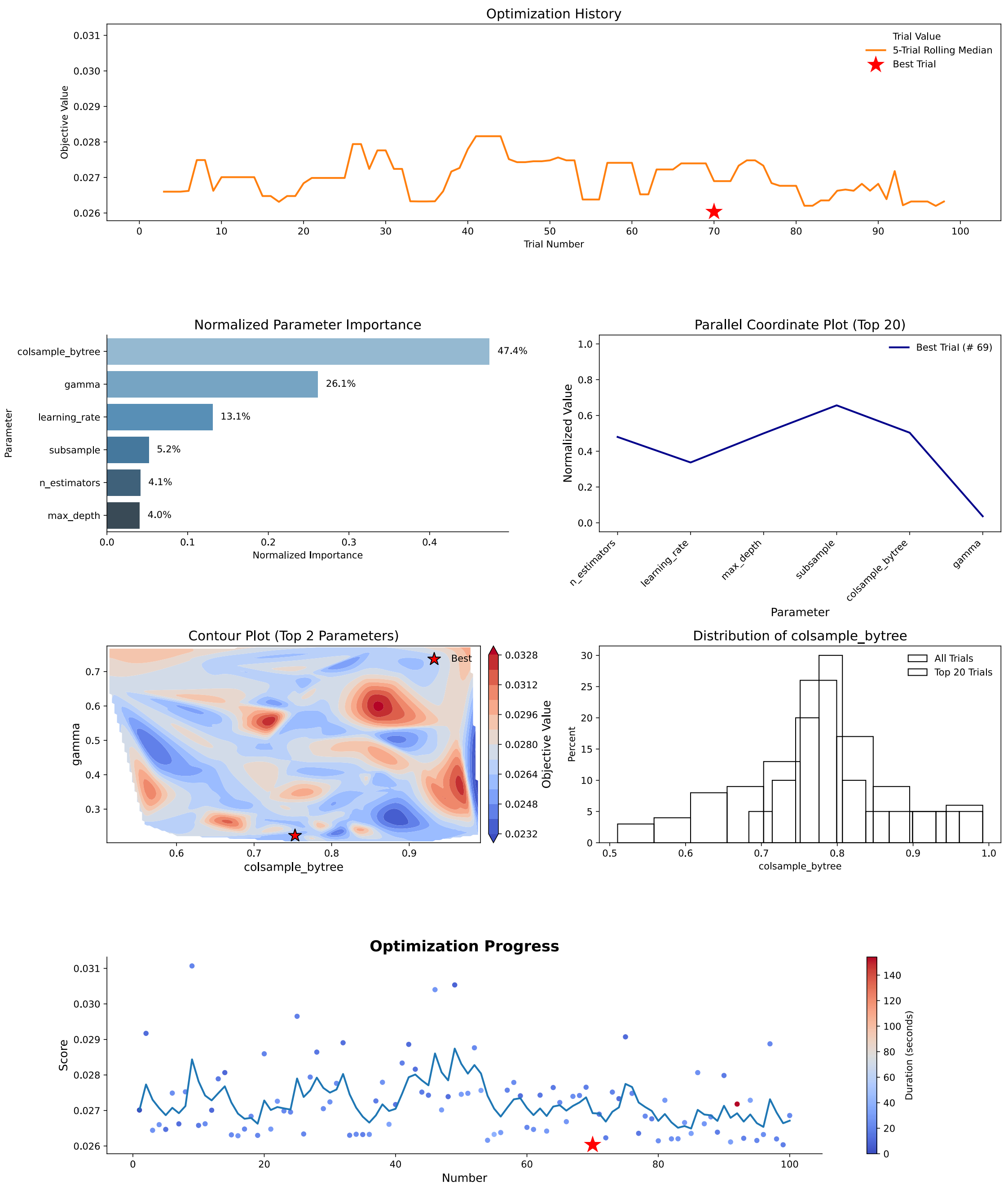


**Figure 4-2: Hyperparameter optimization dashboard generated from 100 Optuna trials for the Gradient Boosting Demand Predictor (GBDP). The figure summarizes key results from the tuning process, including the optimization history (top), normalized parameter importance (top-left), parallel coordinate analysis of top-performing trials (top-right), a contour plot of the two most influential hyperparameters (*colsample_bytree* and *gamma*, bottom-left), and the distribution of the most influential hyperparameter (e.g., *colsample_bytree)* value across all trials (bottom-right). These visualizations highlight convergence patterns, feature sensitivity, and hyperparameter interactions leading to the best-performing configuration (Trial #69)**

## 4.3 Experimental Results

In this section, we evaluate the overall performance of our proposed models across multiple districts and forecasting horizons. The goal is to assess both computational efficiency and predictive accuracy under realistic urban mobility conditions.

Table 4-1 summarizes the training and inference times required for our models across the four forecasting horizons (5, 15, 30, and 60 minutes) in the cities of interest. **Training time** refers to the total duration needed to train a single city-horizon model, while **inference time** denotes the average time taken to produce a single prediction at a specific horizon. The results indicate that the models are highly efficient: training each model typically takes **8 seconds** on average or less in many cases, while inference latency is remarkably low – on average 0.64 **μsec**. This highlights the models' potential for **real-time deployment** in intelligent transportation systems and other urban applications with stringent latency constraints.

**Table 4-1: Training and inference times of the three models across different horizons (avg. of 5 runs per prediction task). Inference std was omitted due to it being negligible compared to the mean value.**

| Dataset (#samples) | Model | Horizon (min) | Train Time (sec) (Full) | Train Time[6] (sec) (2M rows) | Inference Time (μsec) |
|---|---|---|---|---|---|
| Rotterdam (~2.6M) | Regressor | 5 | 13.92 ± 1.01 | 13.68 ± 0.99 | 0.38 |
| | | 15 | 16.52 ± 2.21 | 16.23 ± 2.17 | 0.48 |
| | | 30 | 15.7 ± 0.6 | 15.43 ± 0.59 | 0.43 |
| | | 60 | 25.03 ± 2.13 | 24.59 ± 2.09 | 0.79 |
| | Classifier | 5 | 35.24 ± 0.26 | 34.62 ± 0.25 | 0.38 |
| | | 15 | 44.04 ± 3.98 | 43.26 ± 3.91 | 0.38 |
| | | 30 | 34.79 ± 3.2 | 34.18 ± 3.14 | 0.43 |
| | | 60 | 28.95 ± 0.16 | 28.44 ± 0.16 | 0.79 |
| Amsterdam (~14.3M) | Regressor | 5 | 99.63 ± 5.87 | 87.72 ± 5.17 | 0.42 |
| | | 15 | 91.33 ± 10.84 | 80.42 ± 9.54 | 0.39 |
| | | 30 | 86.58 ± 1.19 | 76.23 ± 1.05 | 0.38 |
| | | 60 | 96.17 ± 4.28 | 84.67 ± 3.77 | 0.44 |
| | Classifier | 5 | 119.2 ± 10.72 | 104.95 ± 9.44 | 0.42 |
| | | 15 | 264.2 ± 37.31 | 232.62 ± 32.85 | 0.39 |
| | | 30 | 261.6 ± 53.52 | 230.34 ± 47.12 | 0.39 |
| | | 60 | 269.06 ± 22.9 | 236.9 ± 20.16 | 0.44 |
| The Hague (~5.8M) | Regressor | 5 | 32.41 ± 0.43 | 30.18 ± 0.4 | 0.36 |
| | | 15 | 31.36 ± 3.19 | 29.2 ± 2.97 | 0.34 |
| | | 30 | 25.29 ± 1.01 | 23.55 ± 0.94 | 0.28 |
| | | 60 | 27.76 ± 1.06 | 25.85 ± 0.98 | 0.30 |
| | Classifier | 5 | 55.72 ± 2.35 | 51.89 ± 2.19 | 0.36 |
| | | 15 | 56.76 ± 1.74 | 52.86 ± 1.62 | 0.34 |
| | | 30 | 79.9 ± 3.78 | 74.4 ± 3.52 | 0.28 |
| | | 60 | 80.95 ± 2.75 | 75.38 ± 2.56 | 0.3 |

Tables 4-2 and 4-3 present the **predictive performance** of our framework, evaluated across the four forecast horizons using both regression and classification metrics. For the **regression task**, we report $R^2$ (coefficient of determination), RMSE (root mean squared error), sMAPE (symmetric mean absolute percentage error) and MAE (mean absolute error), while for the

[6]Train time on the reduced dataset has been estimated considering the computational complexity of the model. XGBoost's time complexity is $O(Kd\|x\|_0 \log n)$ [44]. As a result, to estimate the new training type, the following formula was used $x' \approx x \cdot \frac{\log N'}{\log N}$

**classification task** we evaluate F1-score, accuracy, precision and recall. These metrics collectively measure the accuracy, robustness, and generalizability of the models, with particular sensitivity to the diverse temporal dynamics and population densities across districts.

**Table 4-2: Predictive performance of the regressor across different prediction horizons in the cities of Rotterdam, Amsterdam and Hague**

| City | Horizon (min) | $R^2$ | | | MAE | | | sMAPE | | | RMSE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Q25 | Q50 | Q75 | Q25 | Q50 | Q75 | Q25 | Q50 | Q75 | Q25 | Q50 | Q75 |
| Amsterdam | 5 | 0.95 | 0.96 | 0.97 | 0.77 | 0.97 | 1.22 | 0.03 | 0.03 | 0.05 | 1.05 | 1.29 | 1.61 |
| | 15 | 0.91 | 0.94 | 0.95 | 1 | 1.22 | 1.62 | 0.04 | 0.05 | 0.06 | 1.32 | 1.67 | 2.26 |
| | 30 | 0.87 | 0.9 | 0.92 | 1.12 | 1.52 | 2.11 | 0.05 | 0.06 | 0.07 | 1.53 | 2.09 | 2.86 |
| | 60 | 0.77 | 0.83 | 0.87 | 1.39 | 1.97 | 2.88 | 0.07 | 0.08 | 0.09 | 1.87 | 2.77 | 3.96 |
| Rotterdam | 5 | 0.98 | 0.98 | 0.99 | 1.29 | 4.31 | 5.44 | 0.01 | 0.02 | 0.02 | 1.77 | 5.83 | 7.27 |
| | 15 | 0.95 | 0.97 | 0.97 | 1.58 | 5.47 | 7 | 0.02 | 0.02 | 0.04 | 2.21 | 8.62 | 9.92 |
| | 30 | 0.93 | 0.95 | 0.96 | 2.03 | 7.5 | 10.18 | 0.02 | 0.03 | 0.05 | 2.85 | 10.3 | 14.09 |
| | 60 | 0.9 | 0.93 | 0.94 | 2.3 | 6.54 | 10.13 | 0.02 | 0.03 | 0.05 | 3.22 | 12.35 | 17.78 |
| The Hague | 5 | 0.93 | 0.95 | 0.97 | 0.81 | 1.08 | 1.50 | 0.03 | 0.04 | 0.05 | 1.07 | 1.42 | 1.97 |
| | 15 | 0.91 | 0.93 | 0.95 | 0.89 | 1.29 | 1.69 | 0.04 | 0.04 | 0.06 | 1.21 | 1.71 | 2.23 |
| | 30 | 0.83 | 0.92 | 0.93 | 0.97 | 1.58 | 2.10 | 0.04 | 0.05 | 0.07 | 1.36 | 2.11 | 2.82 |
| | 60 | 0.77 | 0.86 | 0.89 | 1.21 | 1.83 | 2.51 | 0.06 | 0.07 | 0.09 | 1.64 | 2.42 | 3.37 |

**Table 4-3: Predictive performance of the classifier across different prediction horizons in the cities of Rotterdam, Amsterdam and Hague**

| City | Horizon (min) | F1 | | | Acc. | | | Prec. | | | Rec. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Q25 | Q50 | Q75 | Q25 | Q50 | Q75 | Q25 | Q50 | Q75 | Q25 | Q50 | Q75 |
| Amsterdam | 5 | 0.84 | 0.88 | 0.91 | 0.89 | 0.92 | 0.94 | 0.84 | 0.89 | 0.92 | 0.84 | 0.89 | 0.91 |
| | 15 | 0.80 | 0.85 | 0.87 | 0.86 | 0.88 | 0.92 | 0.82 | 0.86 | 0.88 | 0.81 | 0.85 | 0.88 |
| | 30 | 0.76 | 0.80 | 0.83 | 0.81 | 0.85 | 0.89 | 0.77 | 0.80 | 0.84 | 0.77 | 0.82 | 0.84 |
| | 60 | 0.71 | 0.74 | 0.78 | 0.77 | 0.80 | 0.85 | 0.70 | 0.75 | 0.79 | 0.71 | 0.76 | 0.79 |
| Rotterdam | 5 | 0.87 | 0.91 | 0.93 | 0.93 | 0.95 | 0.98 | 0.89 | 0.92 | 0.94 | 0.86 | 0.91 | 0.92 |
| | 15 | 0.83 | 0.90 | 0.91 | 0.91 | 0.93 | 0.98 | 0.86 | 0.90 | 0.92 | 0.81 | 0.88 | 0.90 |
| | 30 | 0.69 | 0.85 | 0.89 | 0.87 | 0.91 | 0.92 | 0.81 | 0.87 | 0.89 | 0.71 | 0.85 | 0.89 |
| | 60 | 0.69 | 0.76 | 0.84 | 0.83 | 0.87 | 0.92 | 0.75 | 0.79 | 0.84 | 0.66 | 0.76 | 0.84 |
| The Hague | 5 | 0.79 | 0.84 | 0.88 | 0.87 | 0.91 | 0.94 | 0.79 | 0.85 | 0.89 | 0.78 | 0.86 | 0.90 |
| | 15 | 0.76 | 0.82 | 0.86 | 0.85 | 0.89 | 0.92 | 0.77 | 0.82 | 0.86 | 0.76 | 0.84 | 0.88 |
| | 30 | 0.73 | 0.78 | 0.83 | 0.83 | 0.86 | 0.90 | 0.74 | 0.78 | 0.83 | 0.73 | 0.81 | 0.85 |
| | 60 | 0.67 | 0.74 | 0.78 | 0.79 | 0.82 | 0.87 | 0.68 | 0.75 | 0.79 | 0.67 | 0.78 | 0.81 |

The models demonstrate strong predictive performance in both cities, particularly for short-term horizons. In Amsterdam, $R^2$ scores remain high across all horizons, with a median of 0.96 at 5 minutes, and 0.83 even at 60 minutes, indicating that the model captures a substantial proportion of the variance in district-level demand. Error metrics such as MAE and RMSE increase gradually with forecast horizon, as expected, but remain within acceptable bounds for operational deployment.

Rotterdam shows exceptionally high $R^2$ values, consistently exceeding 0.95 at all horizons, peaking at 0.98 for the 5-minute horizon. Despite slightly higher absolute error values—reflecting more variability in demand magnitude—this strong $R^2$ performance indicates the model's robustness and reliability across heterogeneous districts. Notably, the sMAPE values for both cities remain low (median values $< 0.08$ across all horizons), underscoring the model's high relative accuracy, especially in capturing proportional changes in demand.

Overall, these results highlight the model's ability to maintain high fidelity in both short- and mid-term forecasts, with particularly stable performance across Rotterdam, and low error propagation over time in both urban contexts.

## 4.4 Feature Importance Analysis

This section presents a comprehensive analysis of feature importance across the three Dutch cities: The Hague, Amsterdam, and Rotterdam. The analysis employs ML-based feature importance metrics to identify the most influential variables in predicting urban dynamics within these metropolitan areas. Feature importance values are presented on a logarithmic scale, ranging from $10^{-10}$ $to$ $10^{-1}$, providing insights into the relative contribution of each variable to model performance.

Feature importance was calculated using ensemble methods that capture both linear and non-linear relationships between predictor variables and target outcomes. The analysis encompasses multiple categories of urban features including crowd dynamics, temporal patterns, network centrality measures, and spatial characteristics. Each feature's importance score represents its contribution to reducing prediction error across the model ensemble

To quantify the above contribution of each feature category to the predictive performance of both the regressor and classifier models, we compute the **average feature importance** across all forecasting horizons (5, 15, 30, 60 minutes). Given the minimal contributions of some features (importance values below $10^{-4}$), the y-axis of the corresponding plots has been log-scaled to improve visualization. The aggregated results are presented in Figures 4-3**,** 4-4 and 4-5 illustrating the feature importance for the regressor. Results are reported separately for **each city**, allowing us to analyze city-specific dynamics and validate the transferability of insights.

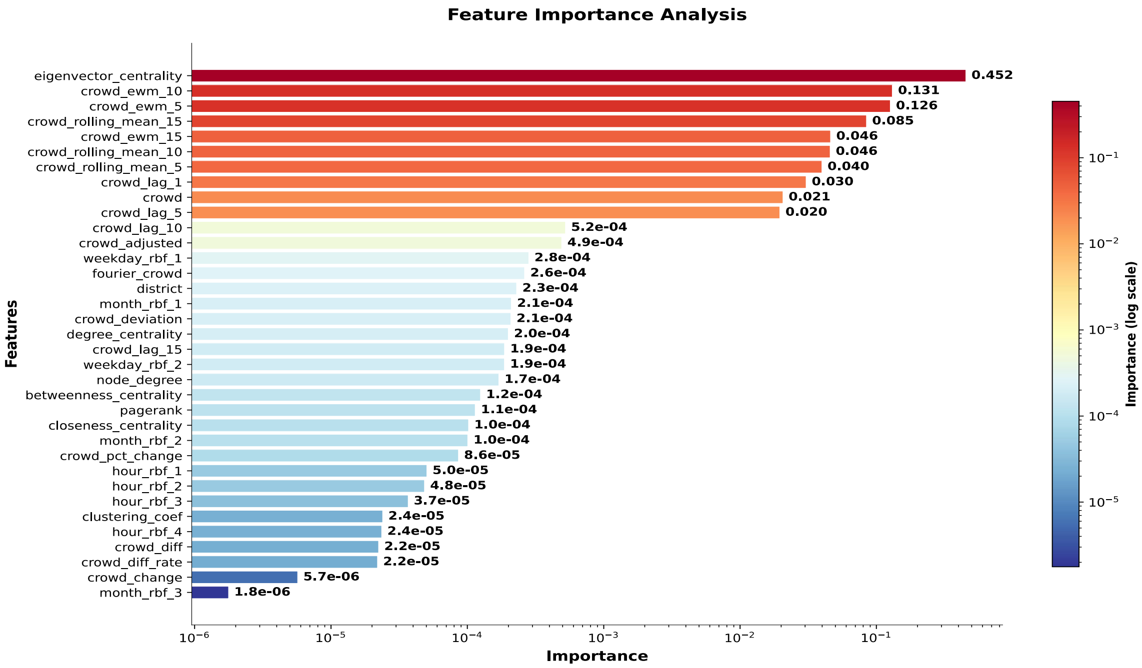


**Figure 4-3: Average feature importance of the regressor across all 4 prediction horizons in the city of Rotterdam**

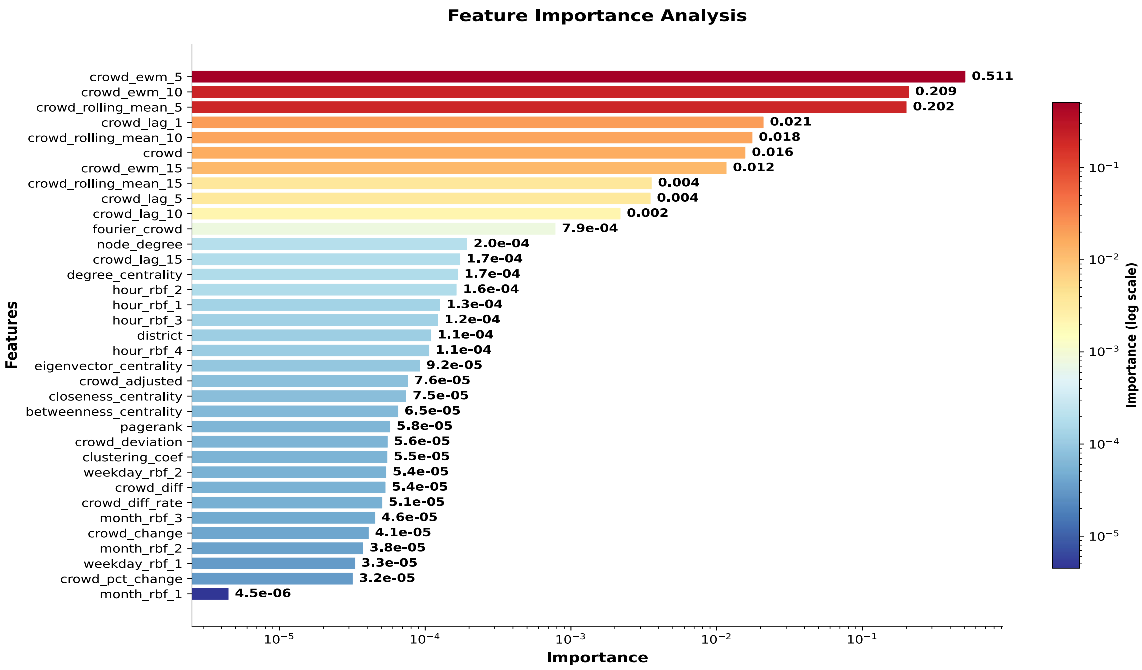


**Figure 4-4: Average feature importance of the regressor across all 4 prediction horizons in the city of Amsterdam**

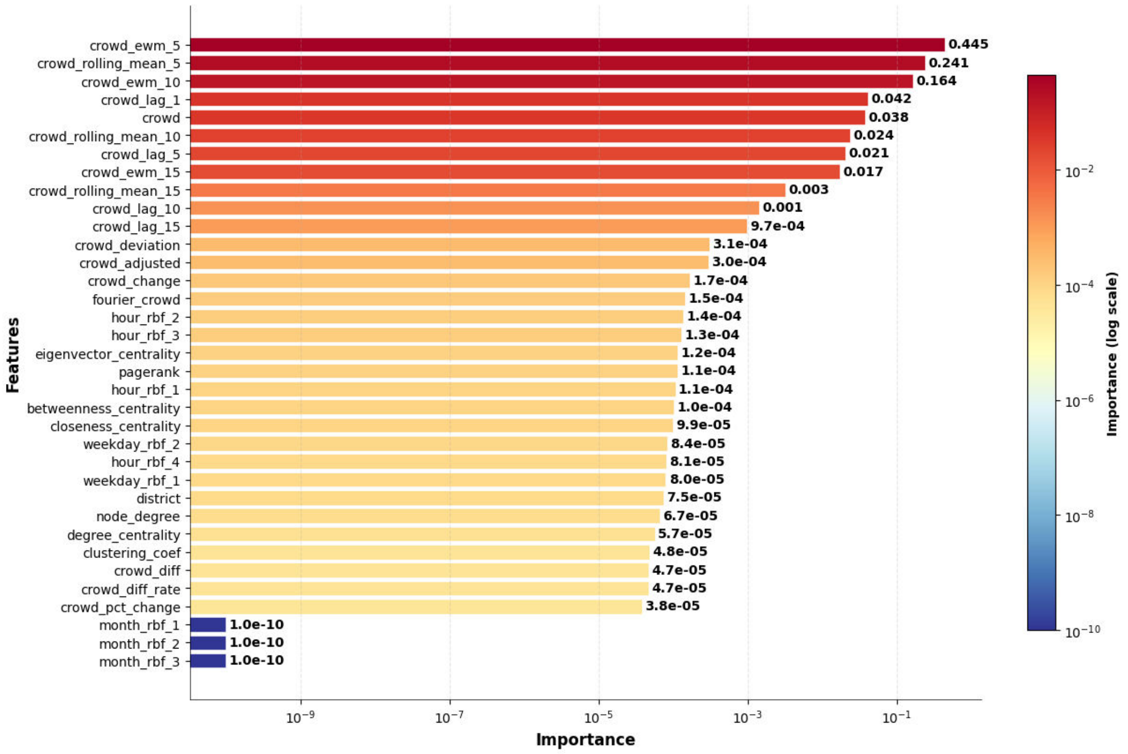


**Figure 4-5: Average feature importance of the regressor across all 4 prediction horizons in the city of Hague**

The *crowd_rolling_mean_5* feature consistently ranks among the top 5 most important variable across all cities. This consistency suggests that short-term crowd dynamics, particularly those captured through exponential weighting and 5-period rolling averages, are fundamental predictors of urban behavior patterns regardless of city-specific characteristics. Crowd lag features demonstrate substantial predictive power across all locations, with the 1-period lag (*crowd_lag_1)* showing importance scores that are on the upper end (between 0.021-0.042). The prominence of lagged crowd variables indicates that temporal dependencies in urban dynamics are crucial for accurate predictions, with immediate past conditions serving as strong indicators of current states. Longer-term crowd patterns, (captured through crowd_rolling_mean_10 and crowd_rolling_mean_15), maintain moderate importance levels (0.003-0.024), suggesting that while immediate temporal patterns dominate, medium-term trends provide additional predictive value.

In contrast, the regressor model for Rotterdam presents a markedly different feature utilization profile. The most important single feature is *eigenvector_centrality*, a graph-based measure of node influence within the city's mobility network. This feature alone contributes 45.2% of the total model importance, underscoring the critical role of structural spatial information in predicting crowd levels. Temporal features such as *crowd_ewm_5* and *crowd_ewm_10* indicate that weighted smoothed recent history still plays a key role, albeit to a lesser extent than in Amsterdam. Furthermore, Rotterdam's model distributes importance more evenly across multiple feature types. Rolling means over longer periods (10 and 15 intervals) are more prominent, and the direct lag features are less influential compared to Amsterdam. This suggests that crowd behavior in Rotterdam is shaped by a combination of local temporal patterns and structural network effects, requiring the model to integrate both types of signals for accurate prediction.

Calendar features exhibit interesting patterns, with hour-based variables (*hour_rbf_1 through hour_rbf_4*) showing moderate importance levels. The relatively low importance of weekday features (*weekday_rbf_1, weekday_rbf_2*) compared to hourly patterns suggests that intra-day variation is more predictive than day-of-week effects in these urban contexts. Notably, monthly features (*month_rbf_1, month_rbf_2, month_rbf_3*) consistently show the lowest importance scores (1.0e-10) across all cities, indicating minimal seasonal variation in the target variable or that seasonal effects are captured through other feature interactions.

A comparative analysis reveals several important distinctions. First, Amsterdam's model is highly concentrated, with the top three features accounting for over 90% of the importance, whereas Rotterdam's is more distributed, with the top three comprising roughly 71%. Second, the reliance on network topology is negligible in Amsterdam but dominant in Rotterdam. This implies a fundamental difference in how crowd flows are governed: **Amsterdam appears to follow regular, short-term patterns**, possibly driven by routine urban behavior, while **Rotterdam's crowd dynamics exhibit stronger spatial dependencies**, possibly influenced by the city's infrastructural layout or transportation network.

Another difference lies in the preference for smoothing techniques. Amsterdam shows a strong bias toward *exponentially weighted moving averages*, while Rotterdam employs a more balanced approach, combining both EWMAs and simple rolling means, particularly over longer horizons. This further supports the hypothesis that Rotterdam's temporal dependencies are more distributed over time, requiring broader aggregation windows to capture relevant trends.

These observations may stem from differences in urban morphology, crowd behavior, or spatial complexity between the two cities. Amsterdam's compact and more homogeneous structure might promote consistent short-term patterns, making it amenable to simple autoregressive models. In contrast, Rotterdam's more heterogeneous layout and distributed activity centers could necessitate the integration of network-aware features to model spatial influence across districts effectively.

In conclusion, this analysis highlights the contextual sensitivity of spatio-temporal forecasting models to the characteristics of the environment in which they are deployed. Understanding such differences is critical for designing location-adaptive models that align with the structural and behavioral patterns of individual cities.

# 5 CONCLUSIONS

This study explored the use of spatio-temporal forecasting techniques for predicting micro-mobility demand in urban environments. By integrating spatial dependencies with temporal dynamics, the proposed methodology effectively captured localized patterns and delivered accurate demand forecasts across multiple time horizons.

Key contributions of this work include the design of a flexible and extensible feature engineering pipeline, which encodes both intra-district and inter-district interactions, and the application of efficient gradient boosting architectures tailored for both regression and classification tasks. This combination yields a highly accurate yet computationally lightweight forecasting framework, suitable for deployment in large-scale, real-time mobility applications.

Experimental results validated the effectiveness of the proposed Gradient Boosted Demand Predictor (GBDP), demonstrating consistently strong performance across cities and horizons. While predictive accuracy naturally declined as the forecast horizon increased, the model maintained robust performance even in longer-range scenarios, highlighting its adaptability and resilience to temporal uncertainty. Beyond the immediate findings, this research opens several avenues for future exploration. One promising direction involves a more systematic study of feature interactions and their contribution to model interpretability and performance. Additionally, incorporating exogenous variables—such as weather conditions, major public events, or transportation infrastructure disruptions—could enrich the feature space and further improve forecast accuracy. Expanding the methodology to cover additional geographic regions or mobility modes would also enable testing the generalizability of the approach under diverse urban conditions.

Finally, to position this work within the broader research landscape, a comprehensive benchmarking study against state-of-the-art spatio-temporal models is planned, under standardized experimental settings. Such an evaluation would provide deeper insight into the relative strengths of the proposed method and guide future refinements.

In conclusion, this thesis demonstrates that data-driven spatio-temporal forecasting methods, when properly designed, can effectively address the challenges of urban mobility prediction. By leveraging the inherent spatial structure of cities and fusing it with high-resolution temporal data, the proposed approach facilitates informed, anticipatory decision-making in smart city planning—ultimately contributing to more efficient, adaptive, and sustainable urban transportation systems [43].

# 6 REFERENCES

[1] Chen, K., Li, G., Li, H., Wang, Y., Wang, W., Liu, Q., & Wang, H. (2024). Quantifying uncertainty: Air quality forecasting based on dynamic spatial-temporal denoising diffusion probabilistic model. Environmental Research, 249, 118438. https://doi.org/10.1016/j.envres.2024.118438

[2] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting (Version 3). arXiv. https://doi.org/10.48550/ARXIV.1707.01926

[3] O'Donncha, F., Hu, Y., Palmes, P., Burke, M., Filgueira, R., & Grant, J. (2022). A spatio-temporal LSTM model to forecast across multiple temporal and spatial scales. Ecological Informatics, 69, 101687. https://doi.org/10.1016/j.ecoinf.2022.101687

[4] Yang, Y., Jin, M., Wen, H., Zhang, C., Liang, Y., Ma, L., Wang, Y., Liu, C., Yang, B., Xu, Z., Bian, J., Pan, S., & Wen, Q. (2024). A Survey on Diffusion Models for Time Series and Spatio-Temporal Data. https://doi.org/10.48550/arXiv.2404.18886

[5] Ye, Z., Kumar, Y. J., Sing, G. O., Song, F., & Wang, J. (2022). A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs. IEEE Access, 10, 75729–75741. https://doi.org/10.1109/ACCESS.2022.3191784

[6] Zheng, C., Fan, X., Wang, C., & Qi, J. (2019). GMAN: A Graph Multi-Attention Network for Traffic Prediction. https://doi.org/10.48550/arXiv.1911.08415

[7] Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 3634–3640. https://doi.org/10.24963/ijcai.2018/505

[8] Lai, Q., Tian, J., Wang, W., & Hu, X. (2023). Spatial-Temporal Attention Graph Convolution Network on Edge Cloud for Traffic Flow Prediction. IEEE Transactions on Intelligent Transportation Systems, 24(4), 4565–4576. https://doi.org/10.1109/TITS.2022.3185503

[9] Jin, M., Wen, Q., Liang, Y., Zhang, C., Xue, S., Wang, X., Zhang, J., Wang, Y., Chen, H., Li, X., Pan, S., Tseng, V. S., Zheng, Y., Chen, L., & Xiong, H. (2023). Large Models for Time Series and Spatio-Temporal Data: A Survey and Outlook. https://doi.org/10.48550/arXiv.2310.10196

[10] Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? Proceedings of the 36th International Conference on Neural Information Processing Systems. Presented at the New Orleans, LA, USA. Red Hook, NY, USA: Curran Associates Inc. https://dl.acm.org/doi/10.5555/3600270.3600307

[11] Duan, P., Mao, G., Yue, W., & Wang, S. (2018). A Unified STARIMA based Model for Short-term Traffic Flow Prediction. 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 1652–1657. https://doi.org/10.1109/ITSC.2018.8569964

[12] Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2014). Traffic Flow Prediction With Big Data: A Deep Learning Approach. IEEE Transactions on Intelligent Transportation Systems, 1–9. https://doi.org/10.1109/TITS.2014.2345663

[13] Sochor, J., Arby, H., Karlsson, I. C. M., & Sarasini, S. (2018). A topological approach to Mobility as a Service: A proposed tool for understanding requirements and effects, and for aiding the integration of societal goals. Research in Transportation Business & Management, 27, 3–14. https://doi.org/10.1016/j.rtbm.2018.12.003

[14] Zhang, J., Zheng, Y., & Qi, D. (2017). Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1). https://doi.org/10.1609/aaai.v31i1.10735

[15] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019, August). Graph WaveNet for deep spatial-temporal graph modeling. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. Presented at the Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}, Macao, China. https://doi.org/10.24963/ijcai.2019/264

[16] Han, S., Mao, H., & Dally, W. J. (2015). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. https://doi.org/10.48550/arXiv.1510.00149

[17] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. https://doi.org/10.48550/arXiv.1503.02531

[18] Nguyen, T., Brandstetter, J., Kapoor, A., Gupta, J. K., & Grover, A. (2023). ClimaX: a25 foundation model for weather and climate. Proceedings of the 40th International Conference on Machine Learning. Honolulu, Hawaii, USA: JMLR.org https://dl.acm.org/doi/10.5555/3618408.3619486

[19] Park, C., Lee, C., Bahng, H., Tae, Y., Jin, S., Kim, K., Ko, S., & Choo, J. (2020). ST-GRAT: A Novel Spatio-temporal Graph Attention Networks for Accurately Forecasting Dynamically Changing Road Speed. Proceedings of the 29th ACM International Conference on Information & Knowledge Management.

[20] Essien, A., Petrounias, I., Sampaio, P. et al. A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. World Wide Web 24, 1345–1368 (2021). https://doi.org/10.1007/s11280-020-00800-3

[21] Essien, A. E., Chukwkelu, G., & Giannetti, C. (2019). A Scalable Deep Convolutional LSTM Neural Network for Large-Scale Urban Traffic Flow Prediction using Recurrence Plots. 2019 IEEE AFRICON, 1–7. https://doi.org/10.1109/AFRICON46755.2019.9134031

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. Proceedings of the 31st International Conference on Neural Information Processing Systems, 6000–6010. https://dl.acm.org/doi/10.5555/3295222.3295349

[23] Zheng, H., Lin, F., Feng, X., & Chen, Y. (2021). A hybrid deep learning model with attention-based Conv-LSTM networks for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 22(11), 6910–6920. https://doi.org/10.1109/tits.2020.2997352

[24] Zhou, T., Niu, P., Wang, X., Sun, L., & Jin, R. (2023). One fits all: power general time series analysis by pretrained LM. Proceedings of the 37th International Conference on Neural Information Processing Systems. Presented at the New Orleans, LA, USA. Red Hook, NY, USA: Curran Associates Inc. https://dl.acm.org/doi/abs/10.5555/3666122.3667999

[25] Jin, K., Wi, J., Lee, E., Kang, S., Kim, S., & Kim, Y. (2021). TrafficBERT: Pre-trained model with large-scale data for long-range traffic flow forecasting. Expert Systems with Applications, 186, 115738. https://doi.org/10.1016/j.eswa.2021.115738

[26] Wang, X., Wang, D., Chen, L., Wang, F.-Y., & Lin, Y. (2023, September 24). Building transportation foundation model via generative graph transformer. 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC). https://doi.org/10.1109/ITSC57777.2023.10422572

[27] Ali, A., Zhu, Y., & Zakarya, M. (2022). Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. Neural networks : the official journal of the International Neural Network Society, 145, 233–247. https://doi.org/10.1016/j.neunet.2021.10.021

[28] Li, Y., Zhao, W., & Fan, H. (2022). A Spatio-Temporal Graph Neural Network Approach for Traffic Flow Prediction. Mathematics, 10(10), 1754. https://doi.org/10.3390/math10101754

[29] Luo, Y., Zheng, J., Wang, X., Tao, Y., & Jiang, X. (2024). GT-LSTM: A spatio-temporal ensemble network for traffic flow prediction. Neural Networks, 171, 251–262. https://doi.org/10.1016/j.neunet.2023.12.016

[30] Guo, J., Huang, W., & Williams, B. M. (2014). Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. Transportation Research Part C: Emerging Technologies, 43, 50–64. https://doi.org/10.1016/j.trc.2014.02.006

[31] Zhang, X., Huang, K., Liu, C., & Xu, X. (2023). Urban Short—Term Traffic Flow Prediction Algorithm Based on CNN-LSTM Model. 2023 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE), 214–217. https://doi.org/10.1109/ICCECE58074.2023.10135384

[32] Wen, H., Lin, Y., Xia, Y., Wan, H., Wen, Q., Zimmermann, R., & Liang, Y. (2023). DiffSTG: Probabilistic Spatio-Temporal Graph Forecasting with Denoising Diffusion Models. Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems, 1–12. https://doi.org/10.1145/3589132.3625614

[33] Lin, L., Shi, D., Han, A., & Gao, J. (2024). SpecSTG: A Fast Spectral Diffusion Framework for Probabilistic Spatio-Temporal Traffic Forecasting. ArXiv, abs/2401.08119.

[34] Mladenović, M. N. (2021). Mobility as a Service. International Encyclopedia of Transportation, 12–18. https://doi.org/10.1016/b978-0-08-102671-7.10607-4

[35] Feng, H. (2021). Predicting the dynamic demand of bike-sharing system in Chicago with Divvy Operation Data. 2021 The 5th International Conference on E-Commerce, E-Business and E-Government, 30–34. https://doi.org/10.1145/3466029.3466035

[36] Xiao, Y., Kong, W., & Liang, Z. (2022). Short-term demand forecasting of urban online car-hailing based on the K-nearest neighbor model. Sensors, 22(23), 9456. https://doi.org/10.3390/s22239456

[37] Sohrabi, S., & Ermagun, A. (2021). Dynamic bike sharing traffic prediction using spatiotemporal pattern detection. Transportation Research Part D: Transport and Environment, 90, 102647. https://doi.org/10.1016/j.trd.2020.102647

[38] Hankey, S., Zhang, W., Le, H. T. K., Hystad, P., & James, P. (2021). Predicting bicycling and walking traffic using Street View imagery and destination data. Transportation Research Part D: Transport and Environment, 90, 102651. https://doi.org/10.1016/j.trd.2020.102651

[39] Le, H. T. K., Buehler, R., & Hankey, S. (2018). Correlates of the built environment and active travel: Evidence from 20 US metropolitan areas. Environmental Health Perspectives, 126(7). https://doi.org/10.1289/ehp3389

[40] Yao, L., Sheng, Q. Z., Ngu, A. H. H., Li, X., Benatallah, B., & Wang, X. (2017). Building entity graphs for the Web of Things Management. Managing the Web of Things, 275–303. https://doi.org/10.1016/b978-0-12-809764-9.00013-5

[41] Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and practice. OTexts.

[42] Zheng, Y., Capra, L., Wolfson, O., & Yang, H. (2014). Urban computing. ACM Transactions on Intelligent Systems and Technology, 5(3), 1–55. https://doi.org/10.1145/2629592

[43] Mokbel, M., Sakr, M., Xiong, L., Züfle, A., Almeida, J., Anderson, T., Aref, W., Andrienko, G., Andrienko, N., Cao, Y., Chawla, S., Cheng, R., Chrysanthis, P., Fei, X., Ghinita, G., Graser, A., Gunopulos, D., Jensen, C. S., Kim, J.-S., … Zimányi, E. (2024). Mobility Data Science: Perspectives and Challenges. ACM Transactions on Spatial Algorithms and Systems, 10(2), 1–35. https://doi.org/10.1145/3652158

[44] Chen, T., & Guestrin, C. (2016, August 13). XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Presented at the KDD ’16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco California USA. https://dl.acm.org/doi/10.1145/2939672.2939785