Urban-AI 2024

Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI

(Urban-AI 2024)

October 29, 2024, Atlanta, Georgia, USA

Editors:

Olufemi A. Omitaomu, Oak Ridge National Laboratory, Tennessee, USA

Ali Mostafavi, Texas A&M University, College Station, Texas, USA

Sukanya Randhawa, Heidelberg University, Germany

Haoran Niu, Oak Ridge National Laboratory, Tennessee, USA

The Association for Computing Machinery, Inc.
1601 Broadway, 10th Floor
New York, NY 10019-7434

ACM COPYRIGHT NOTICE

Notice to Past Authors of ACM-Published Articles

ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written a work that was previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG Newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and where and when published.

ACM ISBN: ISBN:979-8-4007-1156-5

Additional copies may be ordered prepaid from:

ACM Order Department
P.O. BOX 11405
New York, NY 10286-1405
Phone: 1-800-342-6626 (USA and Canada)
+1-212-626-0500 (Global)
Fax: +1-212-944-1318
E-mail: acmhelp@acm.org
Printed in the USA

FOREWORD

The 2nd ACM SIGSPATIAL International Workshop on Advances in Urban AI (Urban-AI 2024) brings together researchers and practitioners to discuss advancements and future directions in urban AI. Urban AI is an emerging field that combines AI, spatial computing, and urban science to address complex challenges faced by cities. The availability of extensive urban data and the growth of digitized city infrastructures have opened opportunities for data-driven machine learning approaches in urban sciences. Urban AI encompasses innovative AI techniques applied to urban problems, AI-ready urban data infrastructure, and various urban applications benefiting from AI. Its applications range from urban planning and design to traffic prediction, energy management, public safety, urban agriculture, and land use.

In the era of digital transformation, cities are becoming smarter, more sustainable, and efficient. Urban AI leverages data collected from sensors, satellites, and IoT devices to enable evidence-based decision making. Real-time analysis of climate patterns, infrastructure performance, energy consumption, and social dynamics helps identify vulnerabilities, optimize resource allocation, and inform climate resilience strategies. Urban AI R&D takes this concept to new heights by utilizing advanced AI algorithms, machine learning, and extensive data analytics to establish intelligent urban ecosystems. These ecosystems integrate various urban domains, fostering cohesive operations, data-informed decision making, and improved citizen experiences. Considering the complexity and heterogeneity of urban information, spatiotemporal aspects such as location, distance, shape, and spatial patterns must be carefully considered and incorporated into urban AI R&D to address geospatial challenges in urban environments.

This year, the workshop received a total of 14 manuscripts. Following a stringent evaluation by the program committee, 10 manuscripts were accepted for presentations—seven manuscripts were accepted as full papers, and three manuscripts were accepted as short papers.

Olufemi A. Omitaomu
Ali Mostafavi
Sukanya Randhawa
Haoran Niu

ACKNOWLEDGEMENTS

We would like to thank the program committee whose reviewing efforts are important for ensuring the quality of the accepted papers.

ORGANIZERS

Olufemi A. Omitaomu, Oak Ridge National Laboratory, Tennessee, USA
Ali Mostafavi, Texas A&M University, College Station, Texas, USA
Sukanya Randhawa, Heidelberg University, Germany
Haoran Niu, Oak Ridge National Laboratory, Tennessee, USA

PROGRAM COMMITTEE

Prof. Chao Fan : Clemson University
Prof. Filip Biljecki : National University of Singapore
Prof. Vanessa Frias-Martinez : University of Maryland at College Park
Dr. Hao Xue : University of New South Wales, Sydney
Prof. Chia-Yu Hsu : Arizona State University
Prof. WenWen Li : Arizona State University
Dr. Majbah Uddin : Oak Ridge National Laboratory
Dr. Abhilasha Saroj : Oak Ridge National Laboratory
Dr. Steffen Knoblauch : Heidelberg Institute for Geoinformation Technology, Heidelberg University, Germany
Dr. Daniela Cialfi : Institute for Complex Systems, Council of National Research of Italy
Prof. Hiba Baround : Vanderbilt University
Prof. Yanjie Fu : Arizona State University

Table of Contents

Session I

Session II

A Graph Deep Learning Model for Station Ridership Prediction in Expanding Metro Networks

Fangyi Ding
The University of Hong Kong
Hong Kong SAR, China
fyding@connect.hku.hk

Yuebing Liang
The Singapore-MIT Alliance for Research and Technology (SMART)
Singapore, Singapore
ybliang@mit.edu

Yamin Wang
The University of Hong Kong
Hong Kong SAR, China
u3603952@connect.hku.hk

Yan Tang
Tongji University
Shanghai, China
2010763@tongji.edu.cn

Yang Zhou
The University of Hong Kong
Hong Kong SAR, China
zhouyhk@hku.hk

Zhan Zhao*
The University of Hong Kong
Hong Kong SAR, China
zhanzhao@hku.hk

Abstract

Due to their reliability, efficiency, and environmental friendliness, metro systems have become a crucial solution to transportation challenges associated with urbanization. Many countries have constructed or expanded their metro networks over the past decades. During the planning stage, accurately predicting station ridership post-expansion, particularly for new stations, is essential to enhance the effectiveness of infrastructure investments. However, station-level metro ridership prediction under expansion scenarios (MRP-E) has not been thoroughly explored, as most advanced models currently focus on short-term predictions. MRP-E presents significant challenges due to the absence of historical data for newly built stations and the dynamic, complex spatiotemporal relationships between stations during expansion phases. In this study, we propose a Metro-specific Multi-Graph Attention Network model (Metro-MGAT) to address these issues. Our model leverages multi-sourced urban context data and network topology information to generate station features. Multi-relation graphs are constructed to capture the spatial correlations between stations, and an attention mechanism is employed to facilitate graph encoding. The model has been evaluated through realistic experiments using multi-year metro ridership data from Shanghai, China. The results validate the superior performance of our approach compared to existing methods, particularly in predicting ridership at new stations.

CCS Concepts

• **Applied computing** → **Transportation**; *Forecasting*.

*Corresponding author

UrbanAI'24, October 29–November 01, 2024, Atlanta, GA

ACM ISBN 979-8-4007-1156-5/24/10
https://doi.org/10.1145/3681780.3697247

Keywords

Ridership Prediction, Metro Expansion, Graph Attention Network, Transport Planning, Urban Development

ACM Reference Format:
Fangyi Ding, Yuebing Liang, Yamin Wang, Yan Tang, Yang Zhou, and Zhan Zhao. 2024. A Graph Deep Learning Model for Station Ridership Prediction in Expanding Metro Networks. In *2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI (UrbanAI'24), October 29-November 1 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3681780.3697247

1 Introduction

Recent decades have witnessed accelerating urbanization, marked by a significant migration from rural to urban areas, especially in developing countries. While urbanization has propelled substantial global advancements, it has also introduced a series of transportation challenges such as traffic congestion and air pollution. Metro systems have emerged as a key solution to these issues, offering a reliable, efficient, and environmentally friendly mode of urban transportation to support burgeoning populations and economic activities. In light of these benefits, metro systems have seen a significant surge in popularity and have flourished in numerous countries globally. During the planning stage of metro expansion, accurately predicting station ridership (e.g., inflow and outflow) after expansion, particularly for new stations, is crucial for enhancing the effectiveness of infrastructure investments and the sustainability of the urban metro system. In this study, we define this task as the Station-Level Metro Ridership Prediction under Expansion Scenario (MRP-E) problem.

Metro network expansion generally includes planning new metro lines or extending existing ones. In the expansion scenario, stations can be divided into three types, as illustrated in Fig. 1. Newly built stations are those not present before the expansion but added to the metro system afterward. Updated stations are those that existed before the expansion but have new lines passing through or extending after the expansion. Existing stations are those that were part of the network before the expansion and have no new lines passing through or extending after the expansion. Among these three types, new stations have no historical ridership data before expansion. Although existing and updated stations have historical ridership

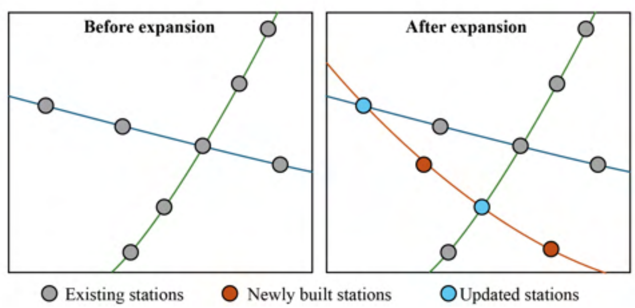


Figure 1: Schematic diagram of station types in the network expansion scenario.

data, their ridership dynamics are affected differently during the expansion (this will be elaborated in Sec. 5.1).

The MRP-E task can be viewed as a crucial precursor to metro network expansion design. Effective expansion design requires accurate demand predictions for new stations during the planning year. While previous studies have made significant contributions to metro network expansion design using optimization models and reinforcement learning algorithms [22, 27, 28, 33], they often overlook the demand prediction step. Many studies either use the current year's all-mode travel demand as a proxy for metro demand in the planning year or rely on unvalidated estimated demands. This approach can compromise the effectiveness of design models, as metro demand is not equivalent to all-mode travel demand, and actual metro demand in the planning year may differ significantly from current ridership figures.

Traditional methods for estimating metro ridership, whether through travel demand modeling [19, 21] or regression models [2, 12], either involve substantial costs and produce approximate estimates, or fail to capture more complex patterns in the data. While significant efforts have been made to apply deep learning methods for metro ridership prediction [4, 29], these methods predominantly target short-term prediction scenarios (e.g., 15 minutes) and assume an unchanged network structure. The potential of artificial intelligence (AI) methods to address the MRP-E problem merits in-depth exploration.

In summary, the MRP-E task is challenging for several reasons. First, under the expansion scenario, the metro network is dynamically evolving, and the spatial interaction between stations can change accordingly. Therefore, deep learning models designed for short-term ridership prediction may not be applicable to the MRP-E task. Secondly, the demand at a given station varies but remains correlated over different times. Previous research primarily focuses on capturing these temporal relationships to forecast future demand. However, these models depend on the availability of historical demand data. In the case of newly built stations within the MRP-E task, where historical data is absent, potential demand must be inferred from surrounding urban contexts and other relevant factors. Third, metro expansion is not a frequent event; new stations and updated stations are much fewer than existing stations, especially in mature metro systems. This limits our ability to obtain sufficient new samples for model training, leading to high performance in predicting existing stations but poor performance in predicting new and updated stations.

In this research, we propose a Metro-specific Multi-Graph Attention Network model (Metro-MGAT) to address the MRP-E problem. Utilizing various urban context data and multi-year metro network data, we extract station-based spatiotemporal features. Then, to capture the complex spatial dependencies, we construct multiple metro-specific correlation graphs. The spatiotemporal features and spatial dependency features are fed into a prediction network to generate predicted station demand. The main contributions of this paper are summarized as follows:

- We develop a Metro-specific Multi-Graph Attention Network model (Metro-MGAT) to address the metro station ridership prediction challenge under expansion scenarios. To the best of our knowledge, this work presents the first attempt to predict expanded metro station demand using deep learning algorithms.
- We design multiple graphs specific to metro systems to encode the complex spatial dependencies among stations. Specifically, a geographical distance graph, a functional similarity graph, and a travel impedance graph are incorporated into the model, and a graph attention mechanism is adopted to facilitate the learning of spatial interaction features.
- To address the class imbalance issue (i.e., new stations are much less numerous than existing stations) in the MRP-E task, we design an age-weighted loss function. This gives "younger" stations, typically those newly built, higher priority, enhancing the model's performance for new stations.

2 Related Works

2.1 Traditional Metro Ridership Prediction Methods

The traditional approach to metro ridership prediction has predominantly utilized four-step methods [11, 19] or activity-based models [1, 21], which are part of the broader travel demand modeling framework. In this framework, metro ridership is viewed as the demand for a specific mode of transportation. While these methods are behaviorally consistent, they require extensive data and a comprehensive calibration of explanatory variables.

As alternatives, several effective models have been proposed to enhance the detail and accuracy of ridership predictions. Time series models, extensively applied in forecasting transportation demand, predict ridership based on regression analyses of past values [7, 34]. Direct ridership models, such as Ordinary Least Squares (OLS) regression and Geographically Weighted Regression (GWR), directly link ridership to accessible factors like local demographics, economic indices, and geographic information [2, 3, 12]. While these statistical methods provide good interpretability, they often struggle to capture the nonlinear dynamics of traffic data and may falter under complex conditions and large datasets.

Additionally, machine learning techniques have been explored for ridership prediction. For example, [26] employed the XGBoost model to predict travel demand in expanding metros. However, existing models have not been deployed on real-world expansion datasets, which may fail to capture the complex spatial interactions between stations during expansion. Consequently, there is a pressing need to develop more accurate models that can realistically represent the planning and deployment of real-world metro system expansions over time.

2.2 Deep Learning Methods in Ridership Prediction

To address the limitations of traditional forecasting methods, some researchers have turned to neural networks (NN) and hybrid models to enhance demand forecasting [20, 24]. Neural networks excel at handling complex nonlinear problems without requiring prior knowledge of the relationships between input and output variables. With the rapid advancements in AI, an increasing number of deep learning algorithms are being utilized to predict metro ridership [17, 29]. Prominent models include Long Short-Term Memory (LSTM) [9] and Gated Recurrent Unit (GRU) [8, 30]. Furthermore, spatial dependency has been integrated into models, employing Convolutional Neural Networks (CNNs) and their hybrid counterparts to explore spatial correlations and achieve network-scale passenger predictions [18, 32].

However, as these CNN-based models often represent metro ridership as grid-based data, there are concerns that this approach may not provide satisfactory accuracy due to the neglect of realistic network topology [4], especially since metro stations are sparsely distributed across urban areas, potentially leading to inadequate capture of representative spatiotemporal patterns [4, 15, 29]. Recently, graph-based models such as the Graph Convolutional Network (GCN) have gained popularity for their ability to consider the topological information of the network [4, 31].

Despite these advancements, most AI applications have focused on short-term prediction scenarios. Over long planning horizons, the evolving nature of metro networks due to expansion can significantly influence ridership distribution, while short-term deep learning prediction models often assume an unchanged network structure. Additionally, although these studies demonstrate the effectiveness of deep learning models for metro demand prediction, they generally rely on sequential/temporal dependencies, which are not applicable for metro expansion scenarios where new stations lack historical data.

Although some recent studies have addressed demand prediction problems in long-term expansion scenarios [14, 16], they predominantly focus on bike-sharing and electric vehicles. Metro expansion differs from these modes in several key ways: 1) the spatial interaction between stations in metro systems is more complex due to the distribution of metro lines; 2) metro expansions are rarer, thus exacerbating the new/existing station imbalance problem; and 3) in metro expansion, there are new/existing stations as well as updated stations, each showing distinct dynamics. To the best of our knowledge, very limited efforts have been devoted to exploring deep learning for demand forecasting in metro expansion scenarios. Given the growing interest in applying deep learning-based methods to future urban and transportation planning, there is a compelling need to further investigate these advanced methods for planning metro systems of the future.

3 Problem Formulation

3.1 Preliminaries

3.1.1 Metro Station Demand. For practical purposes, metro station demand under expansion scenarios does not always necessitate fine-grained forecasting such as hourly or daily levels as is typical in short-term prediction models. Instead, the goal is to capture the total passenger flow at relatively macroscopic time scales for newly planned and updated stations to ensure a reasonable allocation of resources. In this study, to capture the seasonal dynamics, metro station demand is observed at the monthly level. For station i, we aim to predict its demand $d_{i,t}$ at month t as the average daily inflow and outflow. This can be computed as $d_{i,t}^{in/out} = \sum_{m=1}^{M} d_{i,t,m}^{in/out}/M$, where $d_{i,t,m}^{in/out}$ represents the daily inflow and outflow of station i on day m of month t, and M is the total number of days in month t.

3.1.2 Multi-relation Graph. We model the urban metro network as a weighted undirected relational graph. Due to metro expansion, the structure of the metro network evolves dynamically over time, which can be denoted as $G = \{V, E, W, X, D, t\}$, where V refers to the metro stations at month t, E denotes the relationships between stations at month t, W represents the weights of the edges at time t, X indicates the features of stations at month t and D signifies the demand at stations for month t. In this study, we define multiple graphs according to different relationships between stations, which will be introduced in Sec. 4.2.

3.2 Metro Expansion Demand Prediction Problem

Suppose month p marks the transition from the training to the testing period. Given the metro network structure and station features from the earlier training period $\{1, 2, ...p-1\}$, this study aims to develop a mapping function F that correlates station demand with station features, denoted as $D_{i,t}^{in/out} = F(X_{i,t}^{in/out})$. The training period encompasses several metro expansions, enabling this function to capture dynamic mapping relationships under expansion scenarios.

Subsequently, using the anticipated metro network structure and current station features, we apply the mapping function to future expansion scenarios in the testing period $\{p, p+1, ..., P\}$ to generate demand predictions $\hat{D}_{i,t}^{in/out} = F(X_{i,t}^{in/out})$. It is important to note that stations will have their features based solely on the urban environment, network structure, and temporal information, rather than historical demand.

3.3 Framework

Fig. 2 presents an overview of the proposed framework for predicting demand in metro expansion scenarios, which comprises three main components. First, we utilize multi-sourced urban information to extract station-based urban context features. Additionally, temporal factors that influence station demand dynamics are encoded to capture the station-level temporal information. Detailed features will be presented in Sec. 4.1. Second, to capture the spatial correlations between stations, we construct multiple relational graphs to encode various spatial dependency relationships, which

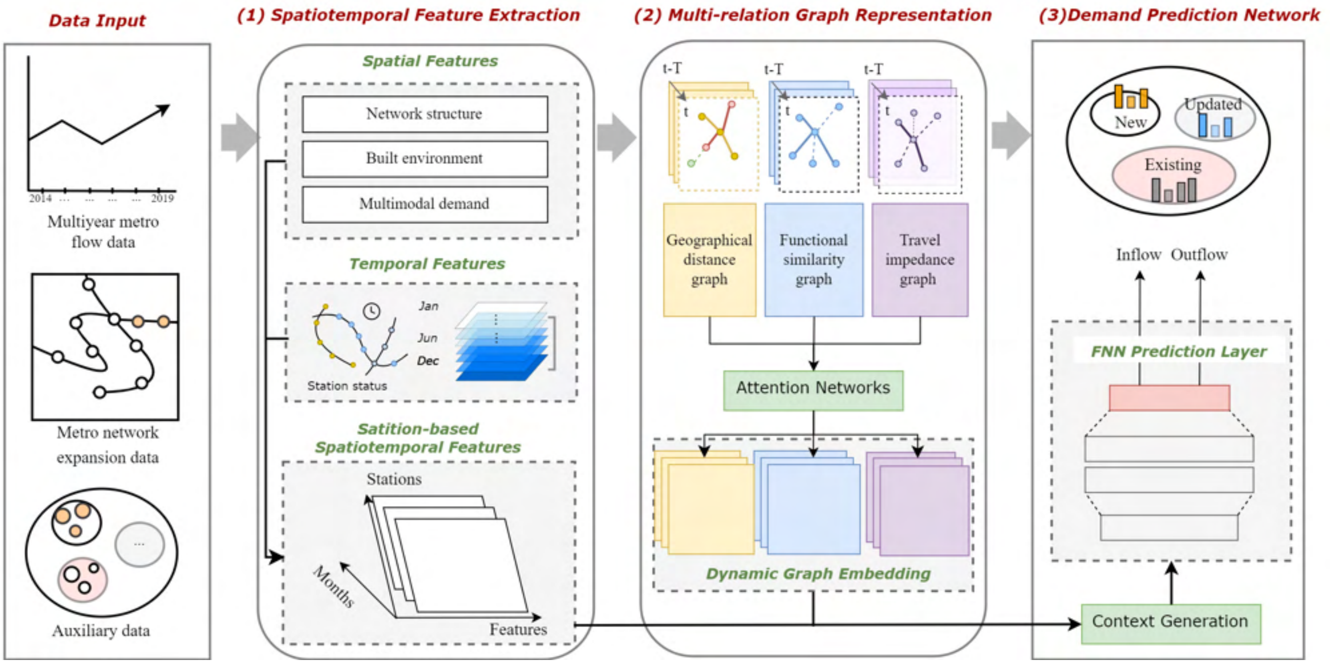


Figure 2: Overview of the proposed framework for expanded demand prediction.

will be detailed in Sec. 4.2. Third, utilizing the spatiotemporal encodings and spatial interaction features, we integrate these features and input them into the prediction network, generating the anticipated demand for newly-built, existing, and updated stations. This component is outlined in Sec. 4.3.

4 Methodology

4.1 Spatiotemporal Feature Extraction

4.1.1 Spatial Features. We utilize diverse urban and metro network information to extract station-based spatial features. These primarily include features related to the metro network structure, the built environment, and multimodal demand. A detailed list of these features is provided in Tab. 1.

Network structure feature are extracted from the metro network's topology, denoted as $x^s_{i,t} \in \mathbb{R}^{12}$. This 12-dimensional vector includes features such as station centrality, the number of connected stations, and the number of connected metro lines. Due to the dynamically evolving topology, we extract these structure features for every observation month t.

Built environment features have been found to be highly correlated with metro flows in previous studies. In this study, we extract a 21-dimensional built environment feature vector $x^b_i \in \mathbb{R}^{21}$. These features, calculated within a 500-meter buffer around stations, primarily include population density [6], house value [1], and the density of 14 categories of Points of Interest (POIs) [2]. Due to data availability, the built environment data is static and was primarily recorded in 2017.

[1]Extracted from *Lianjia*, one of China's largest real estate intermediary companies. https://m.lianjia.com/

[2]Obtained from the Gaode API: https://lbs.amap.com/

Table 1: Summary of features and categories.

Features
Spatial features (dim=37)
1) Network structure (dim=12)
station centralities, metro station connections, metro line connections, interchange station, terminal station…
2) Built environment (dim=21)
population density, house value, POI density of 14 categories, road density, intersection density, landuse diversity…
3) Multimodal demand (dim=4)
taxi pick-ups, taxi drop-offs, bus stop density, bus line density
Temporal features (dim=14)
1) Station status (dim=2)
station age, updated status
2) Month (dim=d_m)
observation month embedding

Multimodal demand can potentially reflect metro demand. For instance, buses and taxis can either compete with or complement metro systems, especially over long distances. Hence, these modes may exhibit complex relationships with metro usage. In this study, we incorporate bus and taxi demand into the model. Taxi demand is calculated as the average daily number of taxi pickups and drop-offs within a 500-meter radius of metro stations. Bus demand is represented by the density of bus stops and bus lines within the same radius. We combine these as the multimodal demand feature vector $x^m_i \in \mathbb{R}^4$, with taxi data specifically collected from Shanghai between April 1st and April 30th, 2015.

Based on the above features, the spatial feature vector for station i at month t can be represented as:

$$x^P_{i,t} = [x^s_{i,t} \parallel x^b_i \parallel x^m_i], \tag{1}$$

where $\parallel$ denotes the concatenation function.

4.1.2 Temporal Features. We include two types of features to encode temporal information: station status and month. The former includes the station's age $x^a_{i,t} \in \mathbb{R}$ (the number of months since it first operated) and update status $x^u_{i,t} \in \mathbb{R}$ (whether it has had new lines pass through within the past year). The latter feature records the month of observation, which is transformed into a month embedding vector $X^m_{i,t} \in \mathbb{R}^{d_m}$, where d_m is the vector dimension. Each month is represented as a unique vector to capture seasonal fluctuations. The temporal features of station i at month t can be represented as:

$$x^T_{i,t} = [x^a_{i,t} \parallel x^u_{i,t} \parallel x^m_{i,t}] \tag{2}$$

4.2 Multi-relation Graph Representation

4.2.1 Multiple Graph Construction. The demand at one station can be influenced by other stations within the metro system. Rather than solely relying on physical topologies or geographic proximity, we construct multiple graphs specifically tailored for metro systems to capture the complex spatial dependencies among stations.

Geographical Distance Graph: Generally, stations that are geographically closer tend to be more correlated. We define a distance graph to model this spatial correlation. In this graph, the geographical proximity between station i and j at month t is computed as:

$$a^g_{ij,t} = \exp(-(\frac{dist_{ij,t}}{\sigma_g})^2), \tag{3}$$

where $a^g_{ij,t}$ is the geographical adjacency weight between station i and j in month t, $dist_{ij,t}$ denotes their geographical distance, and σ_g is set as the standard deviations of $dist_{ij,t}$.

Functional similarity graph: Stations with similar distributions of land use functions may exhibit closer demand patterns. For example, if two stations are located within the same functional zone, the people arriving and departing from these stations are likely to have similar activity patterns, leading to more similar flow distributions. In this study, we use the built environment features surrounding each station to encode their land use function. The function similarity between a pair of stations $\{i, j\}$ is is computed as:

$$a^b_{ij,t} = \exp(-(\frac{Euc(x^b_i, x^b_j)}{\sigma_b})^2), \tag{4}$$

where $a^b_{ij,t}$ is the functional adjacency weight between station i and j in month t, $Euc(x^b_i, x^b_j)$ is the Euclidean distance function, x^b_i and x^b_j are built environment vectors described in Sec. 4.1, and σ_b is set as the standard deviations of $Euc_{x^b_i, x^b_j}$.

Travel impedance graph: For transportation systems such as bikesharing or taxis, travel cost is highly correlated with the distance between two locations. However, in metro systems, transfer times play a critical role in people's travel decision-making. Thus, station pairs with lower travel impedance may exhibit higher spatial interaction. The travel impedance proximity between a station pair $\{i, j\}$ is computed as:

$$a^c_{ij,t} = \exp(-(\frac{Imp_{ij,t}}{\sigma_c})^2), \tag{5}$$

$$Imp_{ij,t} = \min_r\{l^r_{ij,t} + \beta n^r_{ij,t}\}, \tag{6}$$

where $a^c_{ij,t}$ is the impedance-based adjacency weight between station i and j in month t. $Imp_{ij,t}$ is the impedance function, which factors in network distance and transfer times. $l^r_{ij,t}$ is the number of link segments in route r between station pair $\{i, j\}$ at month t, $n^r_{ij,t}$ denotes the number of transfers in route r between station pair $\{i, j\}$ at month t, β is transfer penalty parameter, and σ_c is set as the standard deviations of $Imp_{ij,t}$.

Based on the above three types of adjacency matrices, we select the top-K nearest stations for station i as its neighborhood stations for each category and construct three graphs for each month, denoted as $G^g_{i,t}$, $G^b_{i,t}$, and $G^c_{i,t}$, respectively. Note that for the same station i, its associated graphs may change in different months due to the addition of new stations.

4.2.2 Graph Attention Network Encoding. Based on the constructed graphs $G^g_{i,t}$, $G^b_{i,t}$, and $G^c_{i,t}$, we can extract the spatial features of neighboring stations for each station and fuse them through a graph attention mechanism to generate spatial interaction features $h^g_{i,t}$, $h^b_{i,t}$, and $h^c_{i,t}$. Given the input spatial feature vector of station i $x^P_{i,t}$, the spatial interaction feature is computed as the weighted sum of its neighbors' spatial feature vectors $x^P_{j,t}$. Taking the geographical proximity graph $G^g_{i,t}$ as an example, the geographical interaction feature vector $h^g_{i,t}$ can be represented as:

$$h^g_{i,t} = \textstyle\sum_{j \in N^g_{i,t}} \alpha_{ij,t} W x^P_{j,t}, \tag{7}$$

$$\alpha_{ij,t} = \frac{exp(Attn(Wx^P_{i,t} \parallel Wx^P_{j,t}))}{\sum_{k \in N^g_{i,t}} exp(Attn(Wx^P_{i,t} \parallel Wx^P_{k,t}))}, \tag{8}$$

where $N^g_{i,t}$ denotes the geographical neighborhood station set of station i at month t, W is a shared parameter matrix used to perform a linear transformation on all spatial feature vectors. $\alpha_{ij,t}$ is the normalized attention weight between station i and its neighboring station j at month t, and $Attn(\cdot)$ is a two-layer feed-forward network used to generate pairwise attention scores.

4.3 Demand Prediction Network

4.3.1 FNN Prediction Layer. In the prediction network, the extracted spatiotemporal features along with the learned spatial interaction features are fed into the network, generating the inflow and outflow predictions. In this study, we employ a feed-forward network (FNN) as the prediction layer. Specifically, for a target station i at month t, the prediction network performs the following computations:

$$\begin{aligned} z_1 &= ReLU(W_1([x^P_{i,t} \parallel x^T_{i,t} \parallel h^g_{i,t} \parallel h^b_{i,t} \parallel h^c_{i,t}])), \\ z_2 &= ReLU(W_2 z_1), \\ z_3 &= sigmoid(W_3 z_2), \\ \hat{d}_{i,t} &= W_4 z_3, \end{aligned} \tag{9}$$

where $x^P_{i,t}$, $x^T_{i,t}$, $h^g_{i,t}$, $h^b_{i,t}$, and $h^c_{i,t}$ represent the spatial features, temporal features, and three types of spatial interaction features, respectively. W_1, W_2, W_3, and W_4 are learnable parameter matrices. $\hat{d}_{i,t} \in \mathbb{R}^2$ is the predicted inflow and outflow demand of station i at month t.

4.3.2 Age-weighted Loss Function. Typically, the loss function is designed as the sum of squared errors between the predicted and real-world demand. However, in this model, we prioritize the prediction of newly built stations. Specifically, we design a weighted loss function that adjusts the observation weight based on the station's opening age. The loss function is calculated as follows:

$$\begin{aligned} L_\theta &= \textstyle\sum_{t=1}^{T} \sum_{i=1}^{N_t} \gamma_{i,t} (\hat{d}_{i,t} - d_{i,t})^2, \\ \gamma_{i,t} &= \frac{1}{x^a_{i,t}+1}, \end{aligned} \tag{10}$$

where T is the number of months in the training period, N_t is the number of stations at month t, and $\gamma_{i,t}$ is the sample weight, which is inversely proportional to the station age $x^a_{i,t}$.

5 Experiments

5.1 Data Analysis

In this study, we utilize Shanghai as the case city. The Shanghai metro ridership dataset spans daily passenger inflow and outflow for each station from January 2014 to December 2019. Due to a technical issue, data from April 2014 to October 2014 are missing. Fig. 3 (b) illustrates the number of stations and the average daily ridership for each month from December 2014 to December 2019. It is evident that the Shanghai metro network underwent several expansions during this period, resulting in a general increase in system-wide average ridership. Fig. 3 (a) depicts the network topology in December of 2014, 2017, and 2019. Purple stations represent existing stations, red stations denote newly built stations since the previous stage, and blue stations indicate updated stations that were passed by new lines or received line extensions compared to the previous stage.

The three types of stations exhibit distinct flow dynamics due to the expansion. For example, consider the expansion in December 2015, as illustrated in Fig. 3 (c), where we track the ridership dynamics from January 2015 (one year before expansion) to December 2016 (one year after expansion) for the three types of stations. The flows of 20 sampled existing stations, all new stations, and all updated stations in this expansion are illustrated in this subfigure. The flow at newly built stations showed a significant increasing trend since their opening in December 2015. In contrast, the flow at existing stations remained relatively stable, with only seasonal fluctuations evident. Updated stations, compared to existing ones, displayed more pronounced fluctuations; particularly in December 2015, when the updated stations were first served by new lines or had extensions, some updated stations experienced a noticeable initial surge in ridership followed by a sudden drop. On the other hand, contemporaneous existing stations did not exhibit an increasing trend but rather a decline leading up to the Spring Festival holidays. This suggests that the expansion impacts different types of stations differently: new stations experience a significant and sustained increase in traffic, updated stations show a relatively minor and short-term growth trend, while existing stations are largely unaffected.

5.2 Experiment Settings

In this study, each station-month pair is treated as an individual observation. To capture the dynamics of metro expansion, we utilize data from January 2014 to June 2017 for training and validation, and data from July 2017 to December 2019 for testing. During the training and validation period, 303 stations were operational, while the testing period saw the addition of 31 new stations and updates to 10 stations (i.e., these stations had new lines introduced or line extensions during the testing period).

As discussed in Sec. 5.1, the flow fluctuations exhibit differences between new and updated stations under the influence of expansion. For newly built stations in the testing period, all observations from their months of operation are treated as new observations. In contrast, for updated stations in the testing period, due to their relatively minor fluctuations resulting from the expansion, only observations within one year of the update are considered new observations. All other station-month observations in the testing period are classified as existing observations. Consequently, there are 10,347 observations in the training and validation period, and 9,883 observations in the testing period, which includes 8,960 existing observations and 923 new observations.

5.3 Baseline Models

We evaluate our Metro-MGAT model against five baseline models, which include three non-deep learning models and two advanced deep learning models. The details of these models are as follows:

Linear Regression [25] is a commonly used regression model that assumes linear relationships between metro station demand and influencing factors.

Ridge Regression [10] addresses the issue of multicollinearity in linear regression by incorporating L2 regularization. The regularization strength is controlled by the parameter α, which we set at 0.01.

XGBoost [5] is a tree-based ensemble learning algorithm. It utilizes multiple regression trees in a boosting framework to improve prediction accuracy.

Feed Forward Network (FNN) [23] typically consists of an input layer, several hidden layers, and an output layer. Unlike graph-based models, FNNs do not inherently capture spatial relationships. To ensure a fair comparison, we use the prediction layer from our model as the benchmark for the FNN.

Multi-graph Convolutional Network (MGCN) [13] is designed specifically for graph data. It aggregates features from neighboring stations using predefined weights rather than an attention mechanism. For a fair comparison in this baseline model, we employ the same multiple graphs as those constructed in our Metro-MGAT model.

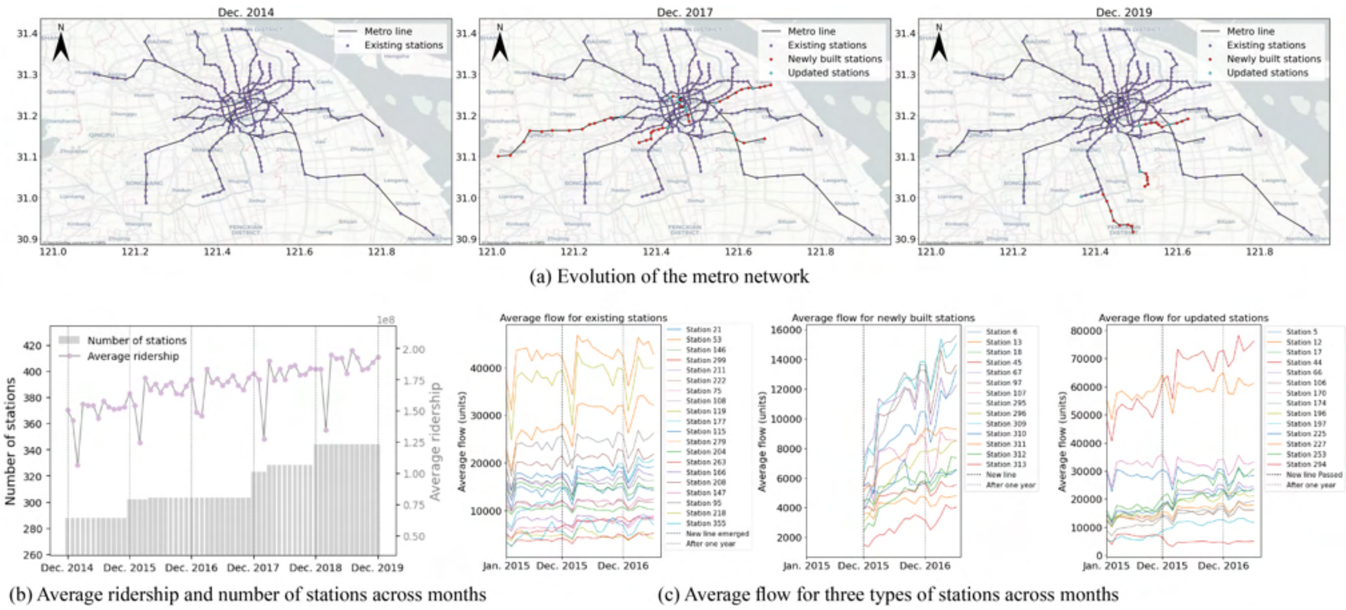


Figure 3: Spatiotemporal analysis of Shanghai metro data.

Spatially-dependent Multi-graph Attention Network (Spatial-MGAT) [14] is a graph neural network approach for predicting the station-level demand in bike-sharing system under expansion scenarios. This model constructs graphs based on geographical proximity and built environment similarity, and uses attention mechanisms to learn the correlation weights between bike stations.

The model performance is evaluated using four metrics: Root Mean Square Error (*RMSE*), Mean Absolute Error (*MAE*), Mean Absolute Percentage Error (*MAPE*), and the Coefficient of Determination (R^2), which are defined as follows:

$$\begin{aligned}
RMSE &= \sqrt{\frac{1}{2\times\sum_{t=1}^{T} N_t}\sum_{t=1}^{T}\sum_{i=1}^{N_t}\sum_{e=0}^{1}(d_{i,t}^{e}-\hat{d}_{i,t}^{e})^2},\\
MAE &= \frac{1}{2\times\sum_{t=1}^{T} N_t}\sum_{t=1}^{T}\sum_{i=1}^{N_t}\sum_{e=0}^{1}\left|d_{i,t}^{e}-\hat{d}_{i,t}^{e}\right|,\\
MAPE &= \frac{100\%}{2\times\sum_{t=1}^{T} N_t}\sum_{t=1}^{T}\sum_{i=1}^{N_t}\sum_{e=0}^{1}\frac{\left|d_{i,t}^{e}-\hat{d}_{i,t}^{e}\right|}{d_{i,t}^{e}},\\
R^2 &= 1-\frac{\sum_{t=1}^{T}\sum_{i=1}^{N_t}\sum_{e=0}^{1}(d_{i,t}^{e}-\hat{d}_{i,t}^{e})^2}{\sum_{t=1}^{T}\sum_{i=1}^{N_t}\sum_{e=0}^{1}(d_{i,t}^{e}-\bar{d})^2},
\end{aligned} \tag{11}$$

where T is the number of months in the test period, N_t is the number of stations in month m, $d_{i,t}^{e}$ and $\hat{d}_{i,t}^{e}$ are the true and predict demand of inflow/outflow (e=1 denotes inflow, and $e = 0$ denotes outflow). $\bar{d}$ is the average demand.

6 Results

6.1 Performance Analysis

In this section, we compare the performance of the Metro-MGAT model with the aforementioned baseline models. For each model, we conduct experiments five times and report the average performance, as listed in Tab. 2. For new observations, our model outperforms all other models across all four metrics. Compared to the second-best model, FNN, our model reduces RMSE, MAE, and MAPE by 7.2%, 6.8% and 8.3%, respectively, while improving R^2 by 4.5%. This improvement may be attributed to our model's enhanced ability to leverage spatial knowledge compared to classical deep neural networks.

Linear regression and ridge regression exhibit similar performance, with XGBoost performing slightly better. Unexpectedly, MGCN performs poorly regarding R^2 in predicting new observations, which may be due to the predefined adjacency weights introducing biased spatial dependencies. In contrast, the graph attention mechanism in Metro-MGAT effectively captures the complex spatial relationships between stations.

For existing observations, XGBoost achieves the best performance across all four metrics, while our model generally ranks third. This is not surprising, as our model prioritizes new observations by assigning higher weights to newly built stations, which can lead to relatively lower performance in predicting existing demand. Prediction models for existing stations have been extensively explored in previous work and have achieved very high accuracy, making them less of a focus in this study.

6.2 Ablation Analysis

In this subsection, we conduct an ablation analysis to verify the effectiveness of key components in our model. Specifically, we create variant models by removing different components and compare their performance with the full Metro-MGAT model. The variant models include: ***-SS:*** Removing station status features, i.e., station age $x_{i,t}^{a}$ and update status $x_{i,t}^{u}$. ***-MD:*** Removing multimodal demand features x_{i}^{m}. ***-WL:*** Using an unweighted loss function, i.e., applying uniform weights to all samples. ***-IG:*** Removing the impedance graph. The performance of these variants is displayed in Tab. 3.

Table 2: Comparison of model performance.

	New observations				Existing observations			
Models	RMSE	MAE	MAPE	R^2	RMSE	MAE	MAPE	R^2
Linear regression	8.471e3	6.226e3	1.381	0.680	8.338e3	6.145e3	0.535	0.768
Ridge regression	8.480e3	6.211e3	1.375	0.678	8.335e3	6.141e3	0.536	0.768
XGBoost	7.712e3	3.833e3	0.536	0.698	***3.789e3***	***2.119e3***	***0.165***	***0.952***
FNN	3.584e3	2.618e3	0.460	0.709	5.710e3	3.857e3	0.305	0.902
MGCN	7.557e3	5.503e3	1.310	0.354	5.689e3	3.894e3	0.348	0.902
Spatial-MGAT	3.829e3	2.894e3	0.473	0.694	5.024e3	3.378e3	0.283	0.923
Metro-MGAT (our model)	***3.325e3***	***2.441e3***	***0.422***	***0.741***	5.068e3	3.449e3	0.323	0.919

Table 3: Performance of variant models.

	New observations				Existing observations			
Models	RMSE	MAE	MAPE	R^2	RMSE	MAE	MAPE	R^2
Metro-MGAT	***3.325e3***	***2.441e3***	***0.422***	***0.741***	5.068e3	3.449e3	0.323	0.919
-SS	3.810e3	2.820e3	0.469	0.631	5.692e3	3.456e3	0.266	0.909
-MD	3.736e3	2.812e3	0.474	0.696	5.390e3	3.696e3	0.338	0.908
-WL	3.565e3	2.691e3	0.438	0.734	***4.643e3***	***3.014e3***	***0.270***	***0.932***
-IG	3.432e3	2.520e3	0.451	0.713	5.657e3	3.735e3	0.347	0.902

The key components of our Metro-MGAT model are shown to be effective in predicting demand for new observations, as indicated by the lower performance across all four metrics when these components are removed. For existing observations, all components, except for the weighted loss function, also contribute significantly to improving model performance, which is expected given that the weighted loss function is specifically designed to prioritize new observations.

7 Conclusion and Discussion

In this study, we propose a Metro-specific Multi-Graph Attention Network (Metro-MGAT) model to address the challenge of station-level metro ridership prediction under expansion scenarios (MRP-E). MRP-E is a complex task due to the absence of historical demand data for newly built stations and the intricate relationship between urban context and station ridership. Additionally, the dynamically evolving network structure and the scarcity of new or updated stations further complicate the problem, rendering recent deep learning models designed for short-term prediction unsuitable. To tackle the MRP-E task, our Metro-MGAT model incorporates three key components. First, we extract station-based spatiotemporal features for each month from various urban data sources and network topology. Next, to capture the spatial dependencies between stations, we construct multiple specialized graphs for metro systems, including geographical distance graphs, functional similarity graphs, and travel impedance graphs. An attention mechanism is employed to effectively capture complex spatial correlations. All features are then fused and fed into a feed-forward network to generate potential metro demand predictions. To address the issue of observation imbalance, we design an age-weighted loss function. To validate our model and capture the complex impacts of network expansion, we conduct experiments on a real-world dataset from the Shanghai metro system, covering the period from 2014 to 2019. The results demonstrate the superiority of our proposed model in predicting demand for new observations and highlight the effectiveness of the model's components.

Additionally, we acknowledge the limitations of our work concerning the adequacy of the experiments and datasets. In the future, we plan to conduct further experiments to differentiate the performance of updated stations from that of newly built stations, and we will utilize additional datasets to validate our model's performance. Furthermore, we aim to enhance our model to develop a demand-driven framework for metro network design.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (42201502) and Seed Fund for Basic Research at The University of Hong Kong (109001117).

References

[1] Muriel Beser and Staffan Algers. 2002. SAMPERS—The new Swedish national travel demand forecasting tool. In *National transport models: Recent developments and prospects*. Springer, 101–118.

[2] Osvaldo Daniel Cardozo, Juan Carlos García-Palomares, and Javier Gutiérrez. 2012. Application of geographically weighted regression to the direct forecasting of transit ridership at station-level. *Applied Geography* 34 (2012), 548–558.

[3] Robert Cervero. 2006. Alternative approaches to modeling the travel-demand impacts of smart growth. *Journal of the American Planning Association* 72, 3 (2006), 285–295.

[4] Pengfei Chen, Xuandi Fu, and Xue Wang. 2021. A graph convolutional stacked bidirectional unidirectional-LSTM neural network for metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems* 23, 7 (2021), 6950–6962.

[5] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[6] CIESIN. 2018. Gridded Population of the World, Version 4 (GPWv4): Population Count, Revision 11. Palisades, New York: NASA Socioeconomic Data and Applications Center (SEDAC). https://doi.org/10.7927/H4JW8BX5. Accessed: day month year.

[7] Chuan Ding, Jinxiao Duan, Yanru Zhang, Xinkai Wu, and Guizhen Yu. 2017. Using an ARIMA-GARCH modeling approach to improve subway short-term ridership forecasting accounting for dynamic volatility. *IEEE Transactions on Intelligent Transportation Systems* 19, 4 (2017), 1054–1064.

[8] Rui Fu, Zuo Zhang, and Li Li. 2016. Using LSTM and GRU neural network methods for traffic flow prediction. In *2016 31st Youth academic annual conference of Chinese association of automation (YAC)*. IEEE, 324–328.

[9] Jianyuan Guo, Zhen Xie, Yong Qin, Limin Jia, and Yaguan Wang. 2019. Short-term abnormal passenger flow prediction based on the fusion of SVR and LSTM. *IEEE Access* 7 (2019), 42946–42955.

[10] Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1 (1970), 55–67.

[11] Goran Jovicic and Christian Overgaard Hansen. 2003. A passenger travel demand model for Copenhagen. *Transportation Research Part A: Policy and Practice* 37, 4 (2003), 333–349.

[12] Myung-Jin Jun, Keechoo Choi, Ji-Eun Jeong, Ki-Hyun Kwon, and Hee-Jae Kim. 2015. Land use characteristics of subway catchment areas and their influence on subway ridership in Seoul. *Journal of transport geography* 48 (2015), 30–40.

[13] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[14] Yuebing Liang, Fangyi Ding, Guan Huang, and Zhan Zhao. 2023. Deep trip generation with graph neural networks for bike sharing system expansion. *Transportation Research Part C: Emerging Technologies* 154 (2023), 104241. https://doi.org/10.1016/j.trc.2023.104241

[15] Yang Liu, Zhiyuan Liu, and Ruo Jia. 2019. DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transportation Research Part C: Emerging Technologies* 101 (2019), 18–34.

[16] Man Luo, Hongkai Wen, Yi Luo, Bowen Du, Konstantin Klemmer, and Hongming Zhu. 2019. Dynamic demand prediction for expanding electric vehicle sharing systems: A graph sequence learning approach. *arXiv preprint arXiv:1903.04051* (2019).

[17] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. 2014. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2014), 865–873.

[18] Xiaolei Ma, Jiyu Zhang, Bowen Du, Chuan Ding, and Leilei Sun. 2018. Parallel architecture of convolutional bi-directional LSTM neural networks for network-wide metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems* 20, 6 (2018), 2278–2288.

[19] Michael G McNally. 2007. The four-step model. In *Handbook of transport modelling*. Vol. 1. Emerald Group Publishing Limited, 35–53.

[20] Brian L Smith and Michael J Demetsky. 1997. Traffic flow forecasting: comparison of modeling approaches. *Journal of transportation engineering* 123, 4 (1997), 261–266.

[21] Laron Smith, Richard Beckman, and Keith Baggerly. 1995. *TRANSIMS: Transportation analysis and simulation system.* Technical Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

[22] Hongyuan Su, Yu Zheng, Jingtao Ding, Depeng Jin, and Yong Li. 2024. MetroGNN: Metro Network Expansion with Reinforcement Learning. In *Companion Proceedings of the ACM on Web Conference 2024*. 650–653.

[23] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. 1997. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems* 39, 1 (1997), 43–62.

[24] JWC Van Lint, SP Hoogendoorn, and Henk J van Zuylen. 2005. Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies* 13, 5-6 (2005), 347–369.

[25] Gerald Walters and Robert Cervero. 2003. Forecasting transit demand in a fast growing corridor: The direct-ridership model approach. *Fehrs and Peers Associates* (2003).

[26] Kaipeng Wang, Pu Wang, Zhiren Huang, Ximan Ling, Fan Zhang, and Anthony Chen. 2022. A Two-Step model for predicting travel demand in expanding subways. *IEEE Transactions on Intelligent Transportation Systems* 23, 10 (2022), 19534–19543.

[27] Lebing Wang, Jian Gang Jin, Gleb Sibul, and Yi Wei. 2023. Designing metro network expansion: Deterministic and robust optimization models. *Networks and Spatial Economics* 23, 1 (2023), 317–347.

[28] Yu Wei, Minjia Mao, Xi Zhao, Jianhua Zou, and Ping An. 2020. City metro network expansion with reinforcement learning. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2646–2656.

[29] Jie Zeng and Jinjun Tang. 2023. Combining knowledge graph into metro passenger flow prediction: A split-attention relational graph convolutional network. *Expert Systems With Applications* 213 (2023), 118790.

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

[31] Jinlei Zhang, Feng Chen, Zhiyong Cui, Yinan Guo, and Yadi Zhu. 2020. Deep learning architecture for short-term passenger flow forecasting in urban rail transit. *IEEE Transactions on Intelligent Transportation Systems* 22, 11 (2020), 7004–7014.

[32] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[33] Liqing Zhang, Leong Hou U, Shaoquan Ni, Dingjun Chen, Zhenning Li, Wenxian Wang, and Weizhi Xian. [n. d.]. City Metro Network Expansion Based on Multi-Objective Reinforcement Learning. ([n. d.]). Available at SSRN: https://ssrn.com/abstract=4837977 or http://dx.doi.org/10.2139/ssrn.4837977.

[34] Yanru Zhang, Yunlong Zhang, and Ali Haghani. 2014. A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. *Transportation Research Part C: Emerging Technologies* 43 (2014), 65–78.

Smart Route: A GIS-Based Solution for Mass Transit Design and Optimization

Siddhant Jakhotiya
IIT Kanpur
Kanpur, Uttar Pradesh, India
siddhantj21@iitk.ac.in

Apratim Sen
University of Calgary
Calgary, Alberta, Canada
apratim.sen@ucalgary.ca

Reza Safarzadeh
University of Calgary
Calgary, Alberta, Canada
reza.safarzadeh@ucalgary.ca

Xin Wang
University of Calgary
Calgary, Alberta, Canada
xcwang@ucalgary.ca

Abstract

Mass transit is a key aspect of urban planning and management. A vast network of mass transit provides various options for connectivity to individuals through extensive networks. On the other hand, a bigger network incurs a huge cost on the operator. Thus, striking a balance between the two is essential and challenging. Furthermore, with the transportation sector being a huge contributor to carbon emission, there is a pressing need to address this environmental impact. This paper presents Smart Route, an app which enables planning and optimization of mass transit for reduction in operating cost and fuel emission while maintaining high levels of passenger satisfaction with the network.

Keywords

Public transit, Mass transit, Bus, Route, Route Planning, Route Optimization, Genetic Algorithm, Simulated Annealing, NSGA-II, Tabu Search

ACM Reference Format:
Siddhant Jakhotiya, Apratim Sen, Reza Safarzadeh, and Xin Wang. 2024. Smart Route: A GIS-Based Solution for Mass Transit Design and Optimization. In *Proceedings of International Workshop on Advances in Urban-AI (Urban-AI '24).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3681780.3697251

1 Introduction

The COVID-19 pandemic resulted in declines in transit ridership in cities worldwide. [11] This increases pressure on operators, who attempt to maintain the quality of service with reduced profits. Thus, cost-cutting in these networks has become a relevant discussion. [10] Additionally, the transportation sector, according to the International Energy Agency [13], accounts for about a quarter of total global carbon emissions and has become the second largest carbon-emitting sector in the world. Thus, reduction in these emissions is of paramount significance in today's age.

In the context of these challenges, planning efficient and sustainable transit networks becomes even more critical. While rail-based systems rely on fixed infrastructure, bus-based networks offer greater flexibility through dynamic rerouting, enabling operators to modify their services according to changing passenger demands. However, this flexibility also introduces complexity, as it leads to a multitude of potential network configurations, each with unique implications for meeting passenger needs and controlling operational costs. Balancing these factors is essential for optimizing bus networks in a post-pandemic world, where both financial sustainability and environmental impact are key considerations.

The planning process for public or mass transit is composed of the following broad steps: resource allocation by selecting bus stop locations, network design using these locations, frequency setting, vehicle scheduling and crew scheduling [8] [15]. The first component (i.e., resource allocation) is essential in determining the effectiveness of the network in terms of the population of the city using it and the next two components (i.e., network design and frequency setting) determine the shape and number of transit routes, route lengths and the number of fleet vehicles based on demand, land use, community characteristics and travel behaviour.

After bus stop locations have been determined and passenger demand between stop locations has been assessed, the problem of planning a bus network is essentially the same as that of the Vehicle Routing Problem with pickup and delivery (VRPPD). [5][7] Capacity constraint can be considered, which is also a standard extension of the Vehicle Routing Problem (VRP). The capacity metric can be expressed in terms of bus seating capacity, which is particularly important when managing a heterogeneous fleet of buses and making decisions on which buses to deploy on specific routes at given times. This combinatorial problem is mathematically intractable. [19] Research in this domain has shown significant promise with the use of heuristics or metaheuristic algorithms. [17]

Optimization of an existing bus network is an equally challenging problem, particularly because there are multiple methods of performing the optimization: removing less busy stops from the network, combining consecutive less busy stops into a single stop which could include relocation/allocation of physical resources to create a new bus stop at a new location, rerouting two routes

Urban-AI '24, October 29, 2024, Atlanta, GA, USA

ACM ISBN 980-1-4503-14424-9/24/10
https://doi.org/10.1145/3681780.3697251

slightly to add an interchange to the network to increase mobility, reducing the frequency of operation along a certain route, amongst many other methods. There is no obvious "best" method for the optimization of a network.

This setting can be viewed as a multi-objective problem, with the objectives of customer satisfaction and the cost of the network being contradictory to each other. It is essential to strike a balance in this trade-off. It can also be treated as a constrained optimization problem, with cost as the only objective and customer satisfaction being the constraint parameters.

While many optimization algorithms exist, there are no approaches to perform planning as well as optimization using the same platform. Our app "`SmartRoute`" performs this integration, where the optimization of bus routes is performed by removing stops along the routes. We present an approach that can enable operators to perform the following tasks: (1) utilize their domain knowledge to iteratively remove stops to optimize the network for cost and visualize its impact, (2) run an algorithm to perform this optimization automatically, (3) draw a route plan based on expected passenger demand. (4) We also provide easy-to-use GIS visualizations and graphs for viewing the effects on costs and passenger satisfaction, measured in terms of walking distance. We provide the operator with a choice amongst three algorithms: Simulated Annealing, Genetic Algorithm and NSGA-II in order to find out the best results across a wide number of potential solutions. We elaborate more on these components in later sections. The algorithm used for route planning is Tabu Search, which works on solving a VRPPD. This problem is relevant in recent times due to increased losses for operators in mass transit and, thus, them implementing cost cutting practices, especially firms that operate private transit networks for their employees.

2 Related Work

Route planning and optimization has been subject to a lot of research over a long period of time, with some reviews already in the literature. Several heuristic as well as metaheuristic algorithms, particularly the Genetic Algorithm, have been utilized in attempts to solve this problem, each presenting their own merits. Ngamchai and Lovell [20] use the Genetic Algorithm for the problem of route design and frequency setting with an aim to maximize transfers. Another study around the same time by Chakroborty [4] performs the route designing and frequency setting task using Genetic Algorithm and argues for why it should be used for the task of Urban Transit Network Design Problem (UTNDP). Another approach using the Genetic Algorithm is presented by Pattnaik et al [21], where they design a network in a two-stage process: first, generate a candidate set of optimal solutions (considering both user and operator cost) and second to find the optimal solution from this set. Both the steps utilize the Genetic Algorithm. Other algorithms have also been tested for in this problem setting. Cancela et al [2] provides a comprehensive approach that uses Mixed Integer Linear Programming (MILP) using a graphical model to design routes from scratch in a public transit network given the stop locations, taking into account the constraints of bus capacity, waiting time and transfers. In more recent work, NSGA-II appears as an enhancement over the traditional Genetic Algorithm since it provides more flexibility, and has been utilized for both, the network design problem [26] as well as network design with frequency setting [3]. Yang et al [26] utilizes NSGA-II whilst considering the number of routes as an important parameter in determining the quality of the network, in terms of directness and travel time for users, and the cost to the operator. Presently, the use of Reinforcement Learning is being explored for this problem [27]. Their work presents a model that takes as input the locations of bus stops and it draws the bus network as routes to maximize the met passenger demand along with the frequency. However, the model only works for a fixed set of stops and needs to be retrained for a different network, thus making it computationally heavy.

The challenge of optimizing existing transit routes has been extensively researched, with numerous studies exploring different optimization methods due to the open-ended nature of the problem. One study [18] demonstrates practical applicability by testing its approach in the city of Dresden, Germany, where it seeks to optimize an existing network by finding a configuration that maximizes passenger demand, by estimating the reaction of existing and potential passengers to the changes in the network. Another study [28] presents an approach to minimize transfers and optimize the route directness, improving passenger satisfaction. They use a greedy algorithm and a fast hill climb approach and compare the effectiveness of these methods.

In particular reference to stop removal and stop spacing, Stewart et al [23] specifically highlights upon the extremely close spacing of stops in North America, originally designed to reduce walking distances for passengers. This creates an inefficiency in the bus network due to longer travel times because of more frequent stops and, additionally, a higher operational cost. Their approach to optimal stop spacing involves formula-based calculation of stop importance. However, there is no utilization of automated algorithms. Another related work [12] explores the Dynamic Programming (DP) paradigm to find optimal stop spacing, but, is based entirely on spacing in terms of distance and the output could require a change in the physical infrastructure present, incurring a cost to the operator.

Our study is motivated by the fact that these two problems, route design and route optimization, are related in practical significance to operators in the present day context of expanding cities with pre-existing networks, along with increasing ecological concerns. We thus aim to solve both of them, using an integrated GIS-based visualization app that can intuitively present the results for the operator, thus demanding minimal technical expertise.

3 Methodology

Our main focus for the app "`SmartRoute`" is cost reduction while maintaining a minimum passenger satisfaction, that is set by the operator. The operator can vary this threshold to view the trends in cost reduction and set it at a value that aligns with their trade-off expectations. The method by which we optimize the bus network is by *stop removal*. For route planning, we take as input the travel demand between every pair of stops and use Tabu Search to find the optimal network, that fulfills all passenger demands.

In this section, we first go over the methods employed for the stop removal module and then the route planning module.

3.1 Stop Removal

Stop removal is processed as follows: for each stop removed, the reduction in the operator cost is computed. Simultaneously, passengers using that stop now have to walk an additional distance to either of the adjacent stops (unless the stop that was removed was a terminal stop - in which case all the passengers move to the single adjacent stop). This results in a decrease in passenger satisfaction. We use this additional walking distance, averaged over all the passengers using that route as the metric to measure the passenger satisfaction and we call it the 'Average Walking Distance'. We treat this as a constrained optimization problem, where the operator provides the threshold for the Average Walking Distance on the route and then the algorithm computes the optimal set of stops that can be removed to minimize the cost, while ensuring that the Average Walking Distance doesn't exceed the provided threshold.

3.1.1 Cost and Passenger Satisfaction. We identified these major sources of cost for the operator:

- Bus purchase cost and annual maintenance cost (AMC)
- AMC for bus stops on the route
- Cost of fuel consumed
- Carbon tax (levied based on the amount of fuel consumed)
- Annual wages for drivers

The annual cost (for a single route) is computed as:

$$\texttt{number of bus trips per day} = \frac{\texttt{operational hours}}{\texttt{route frequency}} \quad (1)$$

$$\begin{aligned}\texttt{driving hours per day} &= \texttt{number of bus trips per day} \\ &\quad \times \texttt{average journey time on route}\end{aligned} \quad (2)$$

$$\begin{aligned}\texttt{annual driver wages} &= \texttt{driver wages per hour} \\ &\quad \times \texttt{driving hours per day} \times 365\end{aligned} \quad (3)$$

$$\texttt{AMC for buses} = \texttt{number of buses in fleet} \times \texttt{AMC per bus} \quad (4)$$

$$\texttt{AMC for stops} = \texttt{number of stops on route} \times \texttt{AMC per stop} \quad (5)$$

$$\begin{aligned}\texttt{fuel cost} &= \texttt{route length} \times \texttt{fuel consumed per km} \\ &\quad \times \texttt{number of bus trips per day} \times 365\end{aligned} \quad (6)$$

$$\begin{aligned}\texttt{carbon tax} &= \texttt{fuel consumed} \\ &\quad \times \texttt{carbon emission per litre of fuel} \\ &\quad \times \texttt{carbon tax per kg of emission} \\ &\quad \times \texttt{number of bus trips per day} \times 365\end{aligned} \quad (7)$$

These calculations give the resultant annual cost under each header. The 'average journey time' denotes the time it takes for a bus to start from one terminal of the route and reach the other terminal. The way this is computed is explained later.

This is a framework for the cost which can very easily be extended to real-world scenarios for operators.

For the purpose of quantifying passenger satisfaction, we choose the metric of additional walking distance for the passenger due to the cost optimization performed by the operator or algorithm. The way this additional walking distance is calculated is as follows: suppose the passenger is currently using some stop number 'x', which the operator or the algorithm decides should be removed from the route. Then the walking distance for this passenger is the distance between this stop number 'x' and either '$x+1$' or '$x-1$'. The method by which the stop is chosen is as follows: suppose the number of passengers using a particular stop number 'x' is n, and the distances between stop number 'x' and '$x-1$' is d_1 and between 'x' and '$x+1$' is d_2. When stop 'x' is removed, passengers using that stop are distributed between the adjacent stop in the inverse ratio of their distance. So, additional passengers using stop number '$x-1$' (beyond the passengers already using that stop) would be $n \times \frac{d_2}{d_1+d_2}$ and those using '$x+1$' would be $n \times \frac{d_1}{d_1+d_2}$. This redistribution is based on a simple and practical assumption of equal density of people using that bus route along the road joining these stops. The average walking distance for a particular bus route would thus be the summation of this additional walking distance for all the passengers divided by the total number of passengers using that bus route. We use this average walking distance as the parameter to measure customer satisfaction. It is also used as a constraint provided by the operator as input to the genetic algorithm, as described later.

3.1.2 Simulated Annealing. Simulated Annealing [22] is an optimization algorithm inspired by the annealing process in metallurgy, where materials are heated and then slowly cooled to reduce defects. The algorithm starts with a high "temperature" allowing for exploration of the solution space by accepting both improvements and deterioration in the solution. As the temperature gradually decreases, the algorithm becomes more selective, increasingly favoring improvements and fewer deterioration. This cooling schedule allows the algorithm to escape local optima and ideally converge to a global optimum solution. The pseudocode for the algorithm is as shown in 1.

The formulation of our problem for the algorithm as follows: we encode the initial route to be an array of 1's, where the length of the array is the same as the number of stops on the route. Each possible solution is a binary array (of the same length), where a 0 denotes that a given stop is now absent from the route and a 1 indicates that the stop is present. In each iteration, we have a current solution, we modify it slightly by choosing a set of random indices in the array and inverting those bits. We then evaluate its fitness i.e. whether it satisfies the constraint provided by the operator for the average walking distance on the route. If it satisfies the constraint, we accept it probabilistically, depending upon the current temperature. After the temperature reaches the stopping temperature, we take the best valid solution (satisfying the constraint) that we've encountered and show its statistics and related GIS visualization on the frontend.

3.1.3 Genetic Algorithm. Genetic Algorithm [16] is a search and optimization technique inspired by the process of natural selection. It uses a population of candidate solutions, which evolve over generations through processes like selection, crossover (recombination), and mutation. The fittest solutions are more likely to be selected to produce offspring, ensuring the gradual improvement of the population. Over time, the algorithm converges towards optimal or near-optimal solutions for a given problem. In contrast to single-solution algorithms like Simulated Annealing and Tabu Search, Genetic Algorithms are population-based algorithms, that

Algorithm 1 Simulated Annealing

Input: Constraint-distance D
Output: Best solution found
Initialize temperature $T \leftarrow T_{\text{initial}}$
Initialize cooling rate α
Generate a random initial solution S_{current} satisfying the constraint D
Set $S_{\text{best}} \leftarrow S_{\text{current}}$
while $T > T_{\text{stop}}$ **do**
  Generate a new solution S_{new} by slightly modifying S_{current}
  if S_{new} is valid **then**
    Calculate the cost C_{current} of S_{current}
    Calculate the cost C_{new} of S_{new}
    if $C_{\text{new}} < C_{\text{current}}$ **then**
      Accept S_{new} as S_{current}
      if $C_{\text{new}} < C_{\text{best}}$ **then**
        Set $S_{\text{best}} \leftarrow S_{\text{new}}$
      end if
    else
      Set $\Delta C \leftarrow C_{\text{new}} - C_{\text{current}}$
      With probability $e^{-\Delta C/T}$, accept S_{new} as S_{current}
    end if
  end if
  Update temperature $T \leftarrow T \times \alpha$
end while
return S_{best}

maintain a set of possible answers and aim to improve upon that set in each iteration.

Our implementation for this algorithm involves a similar encoding for the problem as described before. Each route is converted into an array of length equal to the number of stops. The pseudocode for the algorithm is as show in 2. Here, a chromosome for which the cost of operation is low has a higher fitness, however any chromosome that doesn't satisfy the constraint has 0 fitness. We choose to probabilistically sample based on the fitness instead of selecting the best solutions (elitism) in a generation since this improves the variety in the population and increases the chances of convergence to the global optima before termination of the algorithm [25]. We note that this is a constrained single-objective optimization problem.

3.1.4 NSGA-II. NSGA-II [9] (Non-dominated Sorting Genetic Algorithm II) is a popular multi-objective optimization algorithm designed to find a diverse set of Pareto-optimal solutions [6]. It employs a fast non-dominated sorting approach to classify solutions into different fronts based on dominance, with the first front representing the best non-dominated solutions. NSGA-II uses a crowding distance mechanism to ensure diversity among solutions by measuring the density of solutions surrounding a particular point in the objective space. The algorithm iterates through selection, crossover, and mutation processes to evolve the population, maintaining a balance between exploration and exploitation. Its elitism ensures that the best solutions are carried over to subsequent generations, improving convergence to the true Pareto front.

Algorithm 2 Genetic Algorithm

Input: Constraint-distance D
Output: Best solution found
population ← generate_random_population()
for $i = 1$ to `NUM_ITER` **do**
  Compute fitness for chromosomes in population
  Selection:
    Sample from the population based on fitness of each chromosome to generate the intermediate population
  Crossover:
    Perform crossover to generate offsprings
  Selection:
    Combine offsprings and intermediate population
    Compute combined fitness
    Sample from this combined population
  Mutation:
    Perform mutation to generate the mutated population
  Selection:
    Combine mutated and non-mutated populations
    Compute combined fitness
    Sample the next generation from this combined population
end for
return Best chromosome found

Since NSGA-II is a multi-objective optimization, we have considered two objectives; operating cost and walking distance of the passenger. We simultaneously try to reach an optimum solution for both these metrics. The encoding of the problem for the algorithm remains the same and follows the pseudocode explained in 3.

Algorithm 3 NSGA-II

population ← generate_random_population()
for $i = 1$ to `NUM_ITER` **do**
  Sort the population into fronts
  Selection:
    Compute the crowding distance of all chromosomes in each front
    Sample the intermediate population based on the crowding distance
  Crossover:
    Perform crossover to generate offsprings
  Selection:
    Combine offsprings and intermediate population
    Sort into fronts and compute crowding distance
    Sample from this combined population
  Mutation:
    Perform mutation on the sampled population to generate the mutated population
  Selection:
    Combine mutated and non-mutated populations
    Sort into fronts and compute crowding distance
    Sample next generation from this combined population
end for
return Best chromosome found

3.2 Route Planning

For the purpose of designing a route, we aim to solve a VRPPD. We take as input all predetermined stop locations, as decided by the operator. Using this data, we compute the distance between every pair of stop. Additionally, we also require the expected demand between any pair of stops.

Using this, we find the optimal bus routes network that fulfills all of the passenger demand, subject to a constraint of the maximum length of each route and a maximum number of routes allowed in the network, based on a limit on the number of staff that the operator may have. For this purpose, we use the Tabu Search algorithm, as chosen by the operator when using the platform.

3.2.1 Tabu Search. Tabu Search [14] is a meta-heuristic optimization algorithm designed to solve complex combinatorial problems by exploring the solution space beyond local optimality. Unlike traditional methods, TS utilizes a memory structure, known as a tabu list, to keep track of recently explored solutions or moves, which prevents the algorithm from revisiting them and potentially getting trapped in local optima.

We use this algorithm to solve the VRPPD problem with multiple vehicles to design the routes of the bus network. The strategies we use are: swapping pickup and dropoffs between different vehicles or reordering the pickups-dropoffs within a route. The detailed pseudocode is provided in 4.

Algorithm 4 Tabu Search for Multi-Vehicle VRPPD

Input: Set of pickup and dropoff locations, vehicle fleet size
Output: Best solution found (routes for each vehicle)
Initialize current solution $S \leftarrow$ GenerateInitialSolution()
Initialize best solution $S_{\text{best}} \leftarrow S$
Initialize tabu list $L \leftarrow \emptyset$
for iteration = 1 to max_iterations **do**
  Generate neighborhood solutions by:
    1. Relocating a pickup-dropoff pair to another vehicle's route
    2. Swapping the order of pickup-dropoff pairs within a route
  Evaluate all neighborhood solutions
  Select the best neighborhood solution S_{new} not in tabu list
  if S_{new} is better than S_{best} **then**
    Update $S_{\text{best}} \leftarrow S_{\text{new}}$
  end if
  Update current solution $S \leftarrow S_{\text{new}}$
  Update tabu list L with the move that led to S_{new}
  Remove expired moves from the tabu list
end for
return S_{best}

4 Experiments

For the purpose of building our interface, we use Javascript for the frontend, Python for the backend with Flask for running the server. Additionally, the database is managed using PostgreSQL, where psycopg2 (a Python library) is used to establish communication between the backend server and the database server.

4.1 Data Requirements

For the route planning module, the data requirements are simple: we need only the spatial location of the bus stops and the passenger demand data, i.e. the desired source and destination for each passenger. Using this data, we solve the VRPPD problem using the Tabu search[1] algorithm.

For the optimization module, we require the following data:

- **Routes**: We require a unique key associated with each bus route. Each route is associated with the number of passengers using the route (num_journeys) and the average terminal to terminal time it takes for buses on the route (avg_journey_time). This data is computed using the other tables (passenger transactions and bus schedules respectively). Based on the frequency and the average journey time, the number of bus vehicles required to operate the route are computed (as the fleet_size). Lastly, for the purpose of visualizations, each route is associated with a random color (route_color).
- **Stops**: This is a table which stores all the data associated with each stop on all routes. It stores the stop ID (unique for each stop location). The stop ID, together with the route number act as the primary key for this table – the stop ID alone is insufficient since a given stop can be present on multiple routes. It contains the geographic location of each stop (latitude and longitude) for the purpose of visualization and the passenger traffic witnessed by a stop, in terms of the alighting and boarding passengers. These are computed using the passenger transactions (explained later). Each stop also has a stop number associated with it, which is used for the purpose of creating an ordering of stops, starting from one terminal to the other terminal.
- **Bus Schedules**: This table stores the time at which a bus reaches each stop on a given route. Each entry is associated with a unique bus ID, the route number on which the bus is running and the capacity of each bus. Additionally, each bus is also associated with a direction of travel, where UP denotes that the source of the bus is the terminal with stop number 1 of that route and DOWN denotes the reverse direction of the bus.
- **Passenger Transactions**: This table stores the journey of each passenger, in the form of their boarding and alighting stops along a route. Each passenger is associated with a unique ID (card_code). For each journey, we store the stop ID at which they are going to board the bus (boarding_stop) and the time at which they arrive at this stop (stop_arrival_time), the time at which the bus picks up the passenger (boarding_time), the ID of the stop they wish to get down at (alighting_stop) and the time at which the bus reaches that stop (alighting_time). These times are obtained from the bus schedules table. Additionally, the direction of travel is also stored.

The relationship between all these tables and their attributes are shown in Figure 1.

For the purpose of our project, we use five of the public transit routes and stops from Calgary Transit [24] for simulations. The data

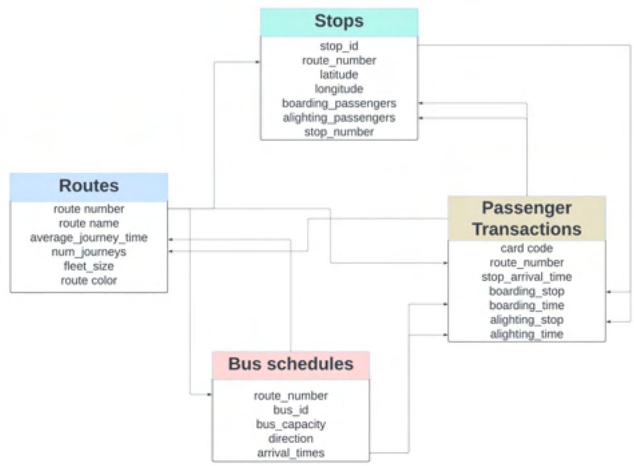


Figure 1: A view showing the relationship between tables and their attributes in the database

present in bus schedules and passenger transactions is randomly generated by us.

For the generation of bus schedules, the frequency is assumed fixed at 30 minutes for all routes. Travel time between two stops is computed, using which the total journey time on a route can be found. We also add a random skew to the travel time between two stops, that ranges between a bus being 2 minutes early upto 3 minutes late. This is aimed at making the data slightly more realistic as the travel time between two stops need not be consistent across multiple journeys due to varying traffic and traffic light conditions.

For the generation of passenger transactions, we randomly choose certain stops along a route to be the source and destination of a passenger. We assign some arbitrary stops as "hotspots" i.e., stops that would witness more passengers travelling to/from it. We also assume that there are more passengers travelling during the hours of 8am-11am and 4pm-8pm. Using these assumptions, we generate 10k transactions, roughly simulating a single day of travel on these three public transit routes on Calgary Transit.

4.2 Stop Removal

4.2.1 Cost Computation. For the computation of cost under various headers (as explained in Section 3.1.1), we assume some constants. The constants used for computation are in Canadian Dollars and are based on realistic estimates in Alberta, Canada. All constants are listed in Table 1.

4.2.2 The App Interface. We have created a very easy-to-use and intuitive interface for the purpose of visualization and decision-making for this problem setting. The dashboard for the same can be seen in Figure 2. The app contains a sidebar on the left, which is divided into different sections. The sidebar enables the operator to interact with the webpage by adding a new route to the database and some viewing options related to existing routes. The snippet in Figure 2 shows the 'Interactive Dashboard', where optimization decisions can be taken by the operator by choosing which stops to remove, using their domain knowledge. On the right side, a map

Type	Value
Driver wages per hour	$20
Annual Maintenance Cost per stop	$500
Annual Maintenance Cost per bus	$5000
Carbon emission	0.025 kg/L fuel
Carbon tax	0.04 $/kg
Fuel cost	1.7 $/L
Operational hours per day	16 hrs/day
Idle time at each stop	0.5 min

Table 1: Constants used for cost computation

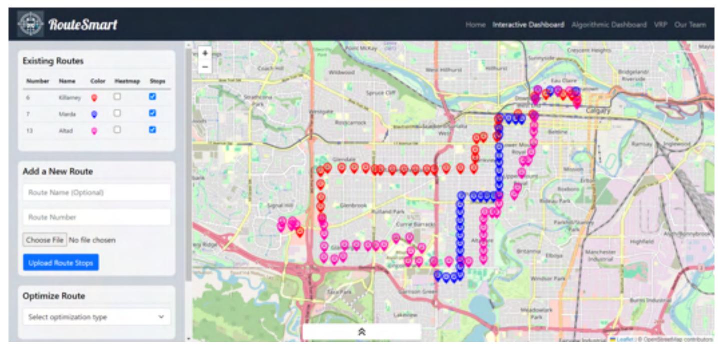


Figure 2: A basic view of the app interface

dominates the majority of the page. Each bus stop location is indicated with a marker. Markers are of different colors to indicate that they belong to different routes, where the color is the same as that of the sidebar and the database. Upon hovering over a particular stop, the operator can view basic statistics related to the stop, for example, the route number to which the stop belongs and its stop number. When the operator clicks on a stop marker, they can see more details about the stop such as how many passengers board/alight at that stop. Additionally, they can view the heatmap of a route by toggling the checkbox on the sidebar. This heatmap is computed using the number of passenger using each stop on the route, and enables the operator to easily identify the hotspots as well as less used stops along each route. The operator can view the effects of their actions using graph visualizations, which can be accessed using an offcanvas at the bottom of the page. The graphs show a comparative analysis between the current and original costs as well as the changes in passenger satisfaction computed in terms of the walking distance along each route, which is initially assumed to be zero (since the original network contains all stops without any removal). These graphs can be seen in Figure 3.

The operator can also view the percentage share in the cost and passenger count of each route as pie charts. This can enable them to realize which routes have a higher contribution to the cost but do not carry a proportionate number of passengers: these can be ideal routes for reducing the cost by removing certain stops. These visual aids are aimed at making the decision making process intuitive. However, this iterative process of making changes and viewing their effects can be time-consuming.

Figure 3: The visualization aid provided for comparing cost v/s customer satisfaction. There have been a few changes made to the network (removal of some stops) and the corresponding cost and walking distance can be observed.

For this purpose, we also provide an 'Algorithmic Interface', where the operator can provide a numeric input for the average walking distance on a route as a constraint and run the desired algorithm (Simulated Annealing or Genetic Algorithm). For running NSGA-II, the operator doesn't provide any constraint distance, since the algorithm in itself works on a multi-objective problem to find a front of "best" solutions.

4.2.3 Results of the Algorithms. We run each of the algorithms with multiple routes in Calgary Transit. With a given constraint distance for optimization, the Genetic Algorithm almost always converges to the same cost, suggesting that it may be the global optima. Note that we run Genetic Algorithm with 300 chromosomes and for 100 generations, however the algorithm usually converges after 40 to 60 generations. For the sake of uniformity, we run NSGA-II with the same parameters. On the other hand, Simulated Annealing converges to different solutions in almost each run, suggesting that it is not able to attain the global minima.

Some metrics measured for each algorithm are listed in Table 2. Route 6 is 34 stops long and Route 13 has 54 stops. Results of other routes are not included since these their trends are similar to the trend presented by these two. As it can be seen, Simulated Anneaing is the fastest, followed by the Genetic Algorithm. For the purpose of comparison with NSGA-II, we select the solution that has the walking distance objective closest to the constraint distance that we'd set for the other algorithms. We've set the average walking distance constraint as $500m$ for both the Genetic Algorithm and Simulated Annealing.

For Simulated Annealing, the number of iterations depend on the initial temperature and the cooling rate. The convergence plot of simulated annealing can be seen in Figure 4. As it can be seen, the algorithm initially has a certain tolerance for accepting worse solutions which decreases over time.

The resultant output by the Genetic Algorithm for one of the routes can be seen in 5. The heatmap layer for the route has also been added to the visualization. As can be seen, the output of the algorithm captures almost all of the hotspots well and reduces stops which are redundant. This sparse spacing of stops reduces the cost of operation significantly, whilst managing the customer satisfaction using the walking distance constraint.

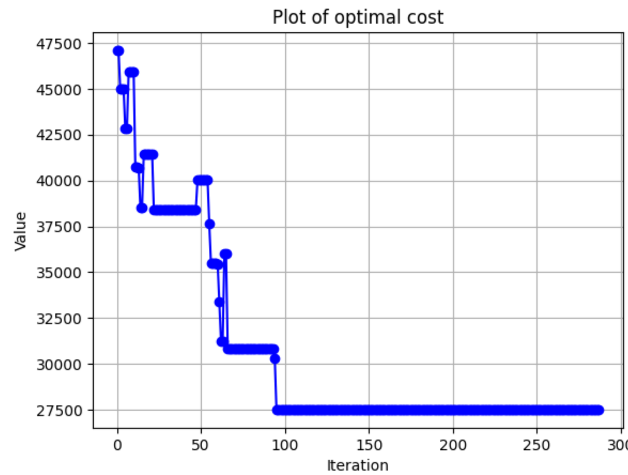


Figure 4: Plot of simulated annealing

Route No	Metric	SA	GA	NSGA-II
6	**Avg. Running Time**	**6**	21	238
	% Reduction in Cost	22.6	**34.4**	24.9
13	**Avg. Running Time**	**9**	32	354
	% Reduction in Cost	21	**37.6**	26.3

Table 2: A table for comparing the algorithms. The route number denotes the bus route number in Calgary Transit. Avg. Running Time refers to the time it takes for the algorithm to output the result and % reduction in cost shows how much operator cost reduction is brought about by the algorithm output.

Figure 5: The output of the Genetic Algorithm. Each black marker denotes stops that are retained as per the output of the algorithm. The heatmap layer on the map shows the relative number of passengers using each stop. Red signifies that the stop experiences a huge number of passengers boarding/alighting every day. As it can be seen, the algorithm captures the hotspots of the route well and also replaces a series of redundant stops with sparsely spaced stops, thereby reducing cost significantly.

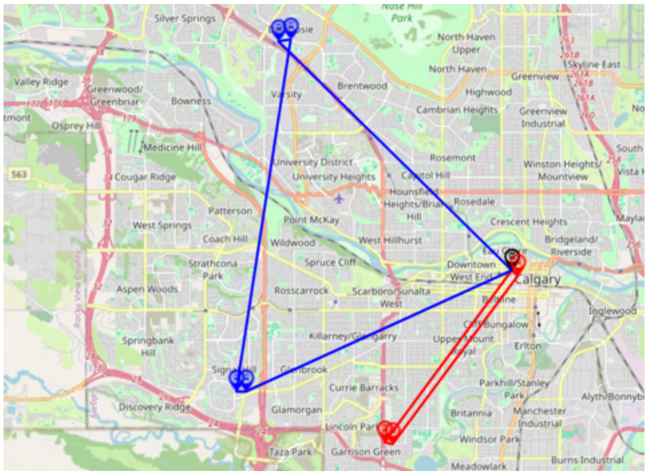


Figure 6: A simple network drawn from the 8 stop locations provided

4.3 Route Planning

For the purpose of route planning, we have a third interface, where the operator can simply upload the locations of bus stops and the pickup-dropoff data (origin-destination matrix), which is essentially the expected passenger demand between a source and destination stop. They can then run the algorithm, which would display the output provided by the Tabu Search algorithm, with each route of the proposed network catering to the passenger demand. Figure 6 shows the sample of a very simple network designed with the given data of passenger demand with the specification of drawing only two routes. The running time for drawing routes is usually 2-3 seconds and is designed to meet the total passenger demand.

5 Conclusion & Future Work

The "SmartRoute" application offers a comprehensive solution to the challenges faced by transit operators in optimizing bus networks. By integrating both route planning and stop removal in a single platform, SmartRoute provides a user-friendly interface that leverages domain expertise and advanced optimization algorithms to strike a balance between cost efficiency and passenger satisfaction. As seen in the experiments, Genetic Algorithm appears as the most effective algorithm for stop removal based optimization. However, NSGA-II has the benefit of flexibility by allowing the user to choose from a series of "best" solutions. Our approach demonstrates the viability of combining automated algorithmic optimization with interactive, GIS-based visualizations, enabling users to make informed decisions.

The results of this study confirm the potential of our app in addressing real-world challenges such as reduced ridership and the need for cost-cutting in mass transit networks, particularly in the post-pandemic landscape. This work is preliminary and offers extensive opportunities for further development. A practical and straightforward enhancement would be to allow the operator to select specific stops on a route that cannot be removed, when running the optimization algorithms. This may be because those stops are interchange stops or have historical significance. Another future work could incorporate both the stop spacing as well as managing the frequency to generate the cost optimal output, ensuring both passenger walking distance and waiting time are minimized in the output. Additionally, there are other potential methods for route optimization, such as relocation of a given bus stop location for better passenger access based on interchanges or proximity to traffic lights. Incorporating such contextual information can enhance the robustness of the outcomes. Moreover, even slight rerouting of combined routes to increase interchanges or better meet passenger demand with this formulation is another potential area for exploration.

Frequency reduction based on the passenger demand would also result in a significant reduction in cost, since a reduced frequency would imply the requirement of a smaller bus fleet and less wages to be paid to bus drivers.

Another method of cost optimization is to use a heterogeneous fleet. We have assumed a fleet of buses having the same carrying capacity, but in practicality, smaller buses are typically used during non-peak hours which in turn, can reduce overall bus maintenance cost and fuel cost. A heterogeneous bus fleet composed of small, medium and large capacity buses, which can be deployed based on traffic, can reduce cost and ensure maximum transit effectiveness. These decisions can be automated using algorithms or artificial intelligence tools.

Transit optimization shall remain a subject of interest for researchers, especially in the light of changing landscapes, evolving demands, ecological and economical concerns, and the increasing focus on "Smart Cities" in the modern era. Additionally, the advancement of machine learning tools, which enable the development of more robust algorithms, further broadens the scope for future research in this area.

Acknowledgments

The authors would like to express their gratitude to everyone who supported this independent study with their insights and encouragement. There was no specific research grant that was utilized.

References

[1] Glass Anderson and Potts. 1997. *Machine scheduling.* Wiley, New York.
[2] Héctor Cancela, Antonio Mauttone, and María E Urquhart. 2015. Mathematical programming formulations for transit network design. *Transportation Research Part B: Methodological* 77 (2015), 17–37.
[3] Shushan Chai and Qinghuai Liang. 2020. An Improved NSGA-II Algorithm for Transit Network Design and Frequency Setting Problem. *Journal of advanced transportation* 2020, 1 (2020), 2895320.
[4] Partha Chakroborty. 2003. Genetic algorithms for optimal urban transit network design. *Computer-Aided Civil and Infrastructure Engineering* 18, 3 (2003), 184–200.
[5] Dantzig and Ramser. 1959. The truck dispatching problem. *Management science* (1959).
[6] Deb and Gupta. 2005. Searching for Robust Pareto-Optimal Solutions in Multi-Objective Optimization. *International Conference on Evolutionary Multi-Criterion Optimization* (2005).
[7] G. Desaulniers, J. Desrosiers, A. Erdmann, M.M. Solomon, and F. Soumis. 2002. *SIAM Monographs on Discrete Mathematics and Application.* SIAM.
[8] Guy Desaulniers and Mark D Hickman. 2007. Public transit. *Handbooks in operations research and management science* 14 (2007), 69–127.
[9] Deb et al. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* (2002).
[10] Kar et al. 2022. Public transit cuts during COVID-19 compound social vulnerability in 22 US cities. *Transportation Research Part D* (2022).
[11] Luyu Liu et al. 2020. The impacts of COVID-19 pandemic on public transit demand in the United States. *Plos One* (2020).

[12] Peter G Furth and Adam B Rahbee. 2000. Optimal bus stop spacing through dynamic programming and geographic modeling. *Transportation Research Record* 1731, 1 (2000), 15–22.

[13] Patrick Gasser. 2020. A review on energy security indices to compare country performances. *Energy Policy* 139 (2020).

[14] Fred Glover. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* (1986).

[15] Valérie Guihaire and Jin-Kao Hao. 2008. Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice* 42, 10 (2008), 1251–1273.

[16] John H. Holland. 1992. Genetic Algorithms. *Scientific American* 267 (1992).

[17] OJ Ibarra-Rojas, L Hernandez, and L Ozuna. 2018. The accessibility vehicle routing problem. *Journal of cleaner production* 172 (2018), 1514–1528.

[18] Michael J Klier and Knut Haase. 2015. Urban public transit network optimization with flexible demand. *Or Spectrum* 37 (2015), 195–215.

[19] Thomas L Magnanti and Richard T Wong. 1984. Network design and transportation planning: Models and algorithms. *Transportation science* 18, 1 (1984), 1–55.

[20] Somnuk Ngamchai and David J Lovell. 2003. Optimal time transfer in bus transit route network design using a genetic algorithm. *Journal of transportation engineering* 129, 5 (2003), 510–521.

[21] SB Pattnaik, S Mohan, and VM Tom. 1998. Urban bus transit route network design using genetic algorithm. *Journal of transportation engineering* 124, 4 (1998), 368–375.

[22] Kirkpatrick S, Gelatt C. D., and Vecchi M. P. 1983. Optimization by Simulated Annealing. *Science, Volume 220, Issue 4598, pp. 671-680* (1983).

[23] Colin Stewart and Ahmed El-Geneidy. 2016. Don't stop just yet! A simple, effective, and socially responsible approach to bus-stop consolidation. *Public Transport* 8 (2016), 1–23.

[24] Calgary Transit and Transit Planning. 2002. A review of bus rapid transit. *Transit Planning* (2002).

[25] Darrell Whitley. 1994. A Genetic Algorithm Tutorial. *Statistics and Computing* (1994).

[26] Jie Yang and Yangsheng Jiang. 2020. Application of Modified NSGA-II to the Transit Network Design Problem. *Journal of Advanced Transportation* 2020, 1 (2020), 3753601.

[27] Sunhyung Yoo, Jinwoo Brian Lee, and Hoon Han. 2023. A Reinforcement Learning approach for bus network design and frequency setting optimisation. *Public Transport* 15, 2 (2023), 503–534.

[28] Fang Zhao and Ike Ubaka. 2004. Transit network optimization–minimizing transfers and optimizing route directness. *Journal of public transportation* 7, 1 (2004), 63–82.

An Advance Review of Urban-AI and Ethical Considerations

Derrick Mirindi†
School of Architecture and Planning
Morgan State University
Baltimore, Maryland, USA
demir1@morgan.edu

David Sinkhonde
Department of Civil Engineering
Pan Africa University Institute for Basic Sciences, Technology and Innovation
Juja, Nairobi, Kenya
sinkhonde.david@students.jkuat.ac.ke

Frederic Mirindi
Faculty of Arts
University of Manitoba
Winnipeg, Manitoba, Canada
mirindif@myumanitoba.ca

Abstract

The rapid digitization of urban infrastructure and the availability of urban data have created opportunities for developing and using artificial intelligence (AI), machine learning (ML), and deep learning (DL) algorithms to address cities' difficult problems. This research offers a thorough evaluation of state-of-the-art AI, ML, and DL algorithms in urban artificial intelligence (AI). This encompasses land use, energy control, public safety, and traffic forecasting. We investigated urban-AI algorithms, and results show that ML algorithms such as Random Forest (RF) can achieve 94% accuracy in urban growth prediction, while Support Vector Machines (SVMs) have demonstrated power in accurately classifying objects such as built-up areas and vegetation. On the other hand, DL algorithms such as Convolutional Neural Networks (CNNs) can attain 79% accuracy in satellite image classification, while Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network are useful in time-series prediction tasks such as traffic flow forecasting, urban air quality prediction, and energy consumption modeling. However, the limitations of these algorithms, particularly when dealing with large datasets, could potentially restrict their scalability in real-time applications. Furthermore, we have identified ethical considerations such as privacy and surveillance, algorithmic bias and fairness, transparency and interpretability, accountability and human oversight, social inclusion, and civic participation, all of which require attention. This has resulted in a variety of suggestions, including creating guidelines for using AI for urban performance to address ethical issues. Future research directions should focus on integrating AI with emerging technologies such as 5G and developing robust frameworks for responsible AI governance in smart cities.

CCS Concepts : **• Computing methodologies → Artificial intelligence → Machine learning.**

Keywords: Urban-AI, artificial intelligence, machine learning, deep learning, ethical considerations

ACM Reference Format:

Derrick Mirindi, David Sinkhonde, and Frederic Mirindi. 2024. An advance review of urban-AI and ethical considerations. In *2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI (UrbanAI'24), October 29-November 1, 2024, Atlanta, GA, USA. ACM, New York, NY, USA, 10 pages.* https://doi.org/10.1145/3681780.3697246

1 Introduction

The rapid urbanization of the 21st century has brought forth unprecedented challenges for cities worldwide [1, 2]. As urban populations continue to grow, reaching an estimated 6.7 billion by 2050 [3], cities face complex issues related to sustainability, resilience, and quality of life [4]. The emergence of smart city initiatives, as well as the increasing availability of urban data, have paved the way for the application of AI in urban contexts to adopt the large and multilayered realities of sustainable implementations [5-7]. Urban-AI, the integration of AI techniques in urban planning, management, and decision-making, holds immense potential to address the multifaceted challenges faced by cities [7–10].

AI, encompassing ML and DL algorithms, has the capability to process vast amounts of urban data, uncover hidden patterns and networks, and provide data-driven

UrbanAI'24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1156-5/24/10
https://doi.org/10.1145/3681780.3697246

insights for urban decision-making [11, 12]. The development of technologies of Internet of Things (IoT) devices and Information and Communication Technologies (ICT) to collect, share urban data sources, and analyze urban data [13-16], has created a rich foundation for AI-driven analysis for optimization and operation [17, 18]. The established workflow led to vast and diverse potential applications in urban planning, transportation management, energy efficiency, and public safety [19].

In the domain of transportation, AI algorithms can optimize traffic flow, reduce congestion, and improve the efficiency of public transit systems [20-22]. Furthermore, energy management can benefit from AI-driven demand forecasting, smart grid optimization, and predictive maintenance of infrastructure [23, 24]. Public safety can be enhanced through AI-powered prediction, emergency response optimization, and crowd management [25]. Environmental monitoring and sustainability efforts can leverage AI for air quality prediction, waste management optimization, and urban green space planning [26]. Moreover, urban-AI can contribute to social equity by identifying and addressing issues of accessibility [27], affordability [28, 29], and inclusion in urban services and amenities [30, 31].

However, the current advance of technology such as sensors and satellites, combined with the development of ML and DL algorithms in AI urban contexts, raises important ethical considerations [32, 35]. To ensure the responsible and equitable deployment of urban-AI, we must address key challenges such as privacy concerns, algorithmic bias, transparency, accountability, and social inclusion. While it is important to explore the application of ML and DL algorithms, it is equally important to cover ethical issues in a systemized and holistic manner, given the rapid and radical rise of technological advancements and their related issues [34, 36]. This study contributes to the understanding of a balanced and ethical usage of AI systems, which remains broad and complex.

The study aims to present a systematic overview of the state-of-the-art AI, ML, and DL algorithms in the field of urban-AI while also exploring its diverse applications, methodologies, limitations, and ethical considerations. Section 2 depicts the method and how we collect data to undertake this research, while section 3 examines the specific AI, ML, and DL algorithms being employed in various urban domains, and section 4 discusses the ethical implications and challenges associated with the integration of AI in urban contexts.

This review contributes to informing policymakers, urban planners, researchers, and practitioners about the opportunities and challenges of integrating AI for sustainable, resilient, and human-centric cities by synthesizing the current research landscape and highlighting the transformative potential of urban-AI. The insights gained from this review can guide future research directions, foster interdisciplinary collaboration, and support the responsible development and deployment of AI in urban environments. Furthermore, this comprehensive review aims to contribute to the growing body of knowledge at the intersection of AI and urban science, as well as the ongoing discourse on leveraging AI to build smarter, more livable, equitable, holistic, and balanced cities.

2 Data and Method

This comprehensive review employs a systematic approach to identify, analyze, and synthesize the current state of research on AI, ML, and DL algorithms in urban contexts. The systematic review methodology ensures a rigorous and transparent process for selecting and evaluating relevant literature and minimizing bias. The review process starts with the formulation of a clear research question, as well as the definition of inclusion and exclusion criteria for study selection. The research question guiding this review is: "What are the current algorithms used, methodologies, and ethical considerations of AI, ML, and DL techniques in urban contexts?" The inclusion criteria encompass peer-reviewed journal articles, conference proceedings, and book chapters published in the English language within the last decade (2010-2024). The focus is on studies that specifically address the application of AI, ML, or DL algorithms in urban planning, management, or decision-making.

We employ a comprehensive search strategy to identify relevant literature, utilizing multiple electronic databases such as Web of Science, Scopus, IEEE Xplore, and Google Scholar. The search query combines keywords related to AI, ML, DL, and urban contexts, such as "artificial intelligence," "machine learning," "deep learning," "urban," "urban planning," and "ethical consideration." We also perform backward and forward citation tracking to find additional relevant studies. We perform data extraction on selected studies, capturing key information such as research objectives, data sources, methodologies, results, and ethical considerations. We synthesize the extracted data using a narrative approach, identifying common themes, trends, and challenges across the reviewed literature. The synthesis also emphasizes the potential of urban-AI in addressing complex urban challenges, as well as the ethical implications of integrating AI in urban decision-making processes.

3 Urban-AI Applications

As shown in Figure 1, ML is a branch of AI where computers interpret and analyze data to create models that can solve problems, while DL is a subset of ML that uses neural networks with multiple layers to improve accuracy and performance. DL is particularly effective in processing unstructured data such as images, audio, and text. Unlike traditional programming, which relies on explicitly coded rules, ML uses data to develop predictive models for making forecasts on new, unseen data.

During the literature review process, we identified a variety of ML algorithms, including Random Forest (RF), Support Vector Machines (SVMs), k-Means Clustering algorithm, Gradient Boosting Machines (e.g., Xgboost), and Time Series Analysis. In addition to that, DL algorithms such as Convolutional Neural Networks (CNNs), Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network, Graph Neural Networks (GNNs), Variational Autoencoders (VAEs) and Transformer Neural Network (TNNs) were also being employed in urban contexts. These algorithms have shown promising results in various urban applications.

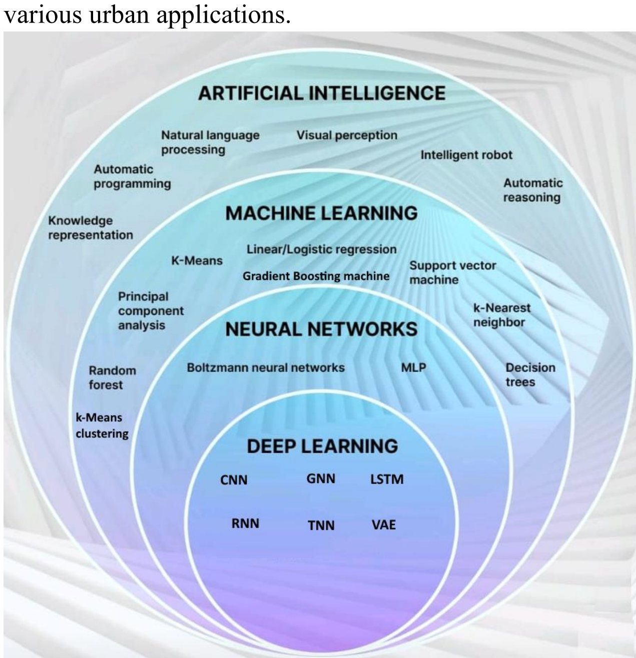


Figure 1. Schematic representation of AI, ML, and DL algorithms in urban context

3.1 Machine Learning Algorithms

3.1.1 Random Forest. RF is a type of ML that uses an ensemble of decision trees to make its prediction. The decision tree can be used for predicting urban growth patterns, land use classification, and traffic flow analysis. We construct it by selecting a random subset of data, such as vegetation, and then build a continuous sequence of decision trees based on these subsets (Figure 2), resulting in a multitude of decision trees. Furthermore, the more decision trees we can utilize, each with a different set of criteria, the better the random forest performs, as it effectively increases prediction accuracy. [37] highlights that RF can achieve 94% accuracy in urban prediction tasks. It is particularly effective in handling complex urban datasets with multiple variables. The general form for a RF prediction can be expressed as:

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} f_b(x) \tag{1}$$

where $\hat{y}$ is the predicted output (e.g., urban growth rate), B is the number of trees (such as 1000 trees), and $f_b(x)$ is the prediction of the *b-th* tree (e.g., the probability of a specific land use type). Studies demonstrated the effectiveness of RF in predicting traffic flow with high accuracy [38]. In addition, according to [39], RF can achieve high accuracy in urban prediction tasks, such as 90.22% in land use classification with a kappa index value of 0.85.

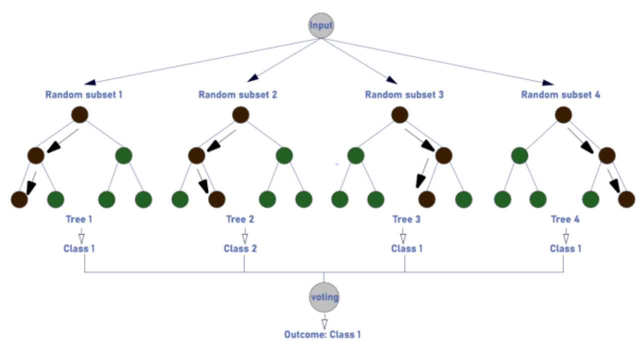


Figure 2. Random forest architecture

3.1.2 Support Vector Machines. SVM is a method for classification of objects (e.g., differentiating between built-up areas and vegetation). This can be done by drawing a hyperplane, 2D line, or 3D plane with all points from one category on one side and all points from another on the other. The hyperplane's goal is to maximize the margin between two categories represented by urban features, such as built-up and non-built-up areas. We can apply SVM to urban image classification, such as identifying building types or land use from satellite imagery. Traffic prediction models also use SVM to predict traffic flow and congestion conditions [40]. A linear SVM uses the following decision function:

$$f(x) = sign(w^T + b) \tag{2}$$

where x is the input vector (e.g., a set of spectral bands from a satellite image), w^T is the weight vector (for example, the coefficients of the hyperplane), and b is the bias term (e.g., a constant that shifts the hyperplane). We can use SMV to classify land use types from satellite imagery, predict flow patterns using historical data from sensors and GPS [41], analyze human mobility patterns using phone application data [42], and extract green urban areas from satellite imagery to facilitate urban planning and environmental management [43, 44]. However, one of the negative aspects of this algorithm is the fact that it can be sensitive to the choice of kernel and parameters, requiring careful tuning to achieve optimal performance [43].

3.1.3 k-Means Clustering algorithm. This algorithm can be utilized for urban segmentation tasks, such as identifying neighborhoods with similar characteristics such as socioeconomic status, land use patterns, and demographic profiles. It can group areas based on energy consumption patterns, helping to optimize energy distribution and sustainable urban infrastructure planning. For instance, a study by [44] demonstrates the wide range of applications of k-means clustering in urban studies, from identifying distinct land use categories to analyzing energy consumption patterns. The objective function for k-Means can be expressed using the Euclidean distance formula below:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \sqrt{\left(x_i^{(j)} - c_j \right)^2} \tag{3}$$

where k is the number of clusters (e.g., 5 clusters representing different urban zones), n is the number of data points (e.g., 1000 data points collected from IoT devices), is the *i-th* data point in cluster j (e.g., the energy consumption pattern of a specific location), and c_j is the centroid of cluster j (e.g., the average energy consumption of a cluster). k-Means Clustering algorithms become important for human mobility because they can analyze movement patterns using data from IoT devices, which helps in the planning of transportation systems and urban infrastructure. It can classify IoT devices based on their usage patterns, which is important for managing and securing urban IoT networks. However, it also has some limitations, such as scalability issues and sensitivity to initial conditions. Indeed, it can be computationally intensive and may not perform well with enormous datasets, which are common in urban IoT applications [45]. Also, the algorithm's performance can be sensitive to the initial placement of centroids, which may lead to suboptimal clustering results.

3.1.4 Gradient Boosting Machines. Gradient boosting machines, particularly Xgboost, are powerful ensemble learning algorithms used in various urban applications. They are employed in predictive maintenance of urban infrastructure, energy demand forecasting, and air quality prediction models. The Xgboost workflow algorithm involves training multiple models on data subsets, focusing on misclassified instances, and aggregating predictions from all models through voting. This process enhances overall accuracy by combining individual model strengths. The general form of the prediction for gradient boosting is:

$$\hat{y}_i = \sum_{m=1}^{M} F_m(x_i) \tag{5}$$

where M is the number of trees in the ensemble (for example, 1000 trees), and Fm is the *m-th* tree in the ensemble. Furthermore, to prevent costly repairs and ensure public safety, Xgboost can identify potential failures in urban infrastructure, such as water pipes and transportation systems [46]. Additionally, we can use it to forecast air quality indices in urban environments, which can inform public health policies and reduce the impact of pollution. It can allow us to classify land use types, such as residential, commercial, and industrial zones, to support urban planning and development. Nevertheless, Xgboost presents some limitations. It can be computationally intensive, especially with large datasets, which may limit its scalability in real-time applications. It can also present an overfitting issue if not properly tuned, leading to poor generalization performance on unseen data. Furthermore, it requires careful tuning of hyperparameters, such as the number of trees and learning rate, to achieve optimal performance [47].

3.1.5 Time Series Analysis. Time series analysis is a critical tool in urban-AI, enabling the forecasting of urban trends such as population growth, traffic patterns, and energy consumption over time. This analysis can be performed using various algorithms, including Autoregressive Integrated Moving Average (ARIMA) and Prophet.

The ARIMA model is a widely used time series forecasting technique that combines three key components:

- Autoregression (AR): This component uses past values of the time series to forecast future values. In urbanism, autoregression can represent historical patterns of urban growth or traffic flow.
- Integration (I): This component involves differentiating the time series to make it stationary, which is essential for accurate forecasting. Integration can help account for seasonal variations in energy consumption or population growth in the urban sector.
- Moving Average (MA): This component uses the errors (residuals) from past forecasts to improve future predictions. Moving average can allow to

capture the random fluctuations in traffic patterns or energy demand in urban-AI.

The ARIMA model can be obtained using the following equation:

$$\left(1-\sum_{i=1}^{p} \varnothing_i L^i\right)(1-L)^d y_t = \left(1+\sum \varnothing_i L^i\right)\varepsilon_t \tag{4}$$

where L, the lag operator, represents the time intervals at which data is collected (e.g., daily, weekly, monthly). In addition, ϕ_i are the parameters of the AR term, which captures the historical patterns in the time series. Also, d is the degree of differencing, θ_i are the parameters of the MA term, which capture the random fluctuations in the time series, and ε_t is white noise. However, some of the disadvantages and limitations include the fact that time series analysis necessitates high-quality data, which can be difficult to obtain in urban contexts, for example. Furthermore, time series data can be non-stationary, which can make it challenging to achieve accurate forecasts.

3.2 Deep Learning Algorithms

3.2.1 Convolutional Neural Networks. Neural networks, known as CNNs, process organized grid data, including images. They employ convolutional layers to learn spatial hierarchies of features from input images in an automated and adaptive manner. Urban image analysis extensively utilizes convolutional layers, which classify satellite images for land use mapping, building identification, and urban transformation detection [48, 49]. According to [50], CNN algorithm can produce an overall accuracy of 79%. The core operation of a CNN can be expressed as:

$$y = f(w \times x + b) \tag{6}$$

where w are the weights, representing the importance of different features in the image, x is the input image, such as a satellite image, and b is the bias term, which shifts the activation function.

3.2.2 Recurrent Neural Network and Long Short-Term Memory network. They are applied in time-series prediction tasks such as traffic flow forecasting, urban air quality prediction, and energy consumption modeling [51-53].

$$h_t = f\left(w_h h_{h-1} + w_x x_t + b\right) \tag{7}$$

where h_t is the hidden state at time t, f is indeed the activation function, typically tanh or ReLU, which introduces non-linearity into the network, w_h is the weight matrix for the hidden state from the previous time step, w_x is the weight matrix for the current input, determining how much influence the current input has on the hidden state, and b is the bias term, which shifts the activation function. RNNs and LSTMs in traffic flow forecasting can predict traffic flow patterns based on historical data, aiding in traffic management optimization, congestion reduction, and temporal dependencies in time-series data, making them suitable for predicting future values. Additionally, they are applicable to a variety of time-series prediction tasks such as forecasting traffic flow and air quality.

3.2.3 Graph Neural Networks. Indeed, the graph simply consists of nodes, or vertices, and connections between these nodes, which are called edges. Urban networks, including transportation systems, social networks, and utility grids, utilize this technique for routing optimization, interaction prediction, and anomaly detection [54].

$$H_r^{(k)} = aggreagate\left(Neighborhood\left(H_r^{(k-1)}\right)\right) \tag{8}$$

Where $H_r^{(k)}$ is the representation of node v at layer k. The aggregate $\left(Neighborhood\left(H_r^{(k-1)}\right)\right)$ function is a key component of GNNs. It combines information from a node's neighbors to update its representation. This process can involve, for instance, summing or averaging neighbor representations, applying learned weights to neighbor representations, and using more complex functions like attention mechanisms. GNNs can be applied to large-scale urban networks, making them suitable for analyzing complex systems. However, they require high-quality data to train effectively, which can be challenging to obtain in urban contexts.

3.2.4 Variational Autoencoders. We use VAEs to generate new data samples and improve design optimization. VAEs are generative models that learn to encode input data into a latent space and then decode it back to the original space. They introduce a probabilistic approach to encoding, which allows for the generation of new data samples. A VAE can be used for dimensionality reduction, feature learning, and anomaly detection in urban data sets, such as identifying unusual patterns in traffic or energy consumption. VAEs can be challenging to train, requiring careful tuning of hyperparameters. We obtain the loss function of a VAE as follows:

$$\boldsymbol{\eta} = E_{q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)\,|p(z)) \tag{9}$$

where: $q(z|x)$ is the encoder, which maps the input data to a latent space; $p(x|z)$ is the decoder, which maps the latent space back to the original space; D_{KL} is the Kullback-Leibler divergence, which measures the difference between the encoder and decoder distributions [55].

3.2.5 Transformer Neural Network. Urban planning-related Natural Language Processing (NLP) tasks, such as analyzing citizen feedback or processing urban policy documents, increasingly use TNNs for spatial-temporal prediction tasks in urban environments. The transformer consists of six encoders on the left and six decoders on the right. Each encoder has a feed-forward layer and a multi-head attention layer, while the decoder has two masked multi-head attention layers and a feed-forward layer (Figure 3) [54]. We define the core attention mechanism in transformers as follows:

$$A(Q, K, V) = softmax\left(\frac{Q\,k^{T}}{\sqrt{d_k}}\right) \tag{10}$$

where Q, K, and V are query, key, and value matrices respectively, and d_k is the dimension of the key vectors.

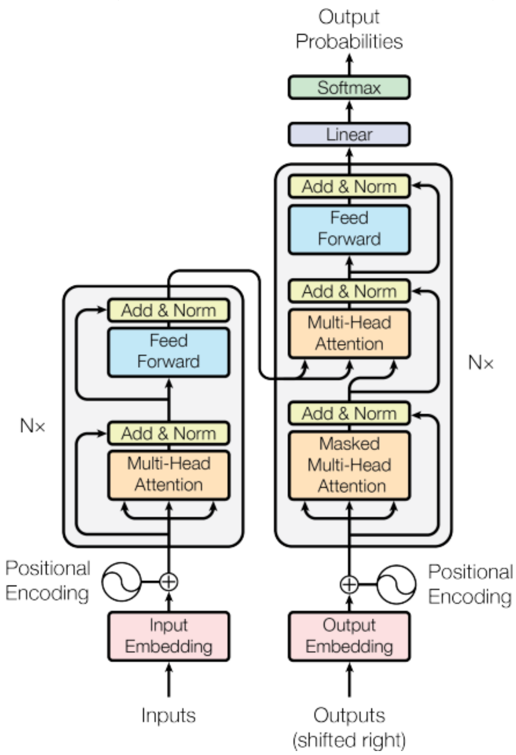


Figure 3. Model architecture of the Transformer [56]

4 Ethical and Legal Issues of AI in Urban

The use of AI in cities brings with it a series of ethical issues that must be carefully addressed to ensure that urban areas are not only smart and intelligent but also responsible. As cities increasingly deploy AI to streamline operations, enhance services, and improve quality of life, it is important to consider the potential risks and challenges. This section explores five main ethical issues when incorporating AI in urban contexts and provides possible strategies and tactics to overcome these issues. Considering those strategies, they regulate and structure a balanced and responsible working scheme for governments, AI developers, investors, and other stakeholders incorporating AI systems.

4.1 Privacy and Surveillance

Smart city solutions, such as sensing devices, IoT devices, and facial recognition systems, can gather massive amounts of individual data [57]. While this data can be used to provide better public services and safety [58], it also increases privacy vulnerabilities through unauthorized access or misuse [59, 60]. AI surveillance systems in public spaces have already been used to track individuals without their consent [34, 35], and sensitive attributes such as ethnicity, habits, or socioeconomic status can be inferred from seemingly innocuous data points by predictive algorithms. It is critical that these collected data participate in automated decision-making processes that end in regulating people's lives [34]. Balancing the use of data for public good with the protection of privacy rights is a critical challenge in urban-AI deployment.

To address these concerns, cities need strong data governance frameworks and privacy-preserving technologies, including rigorous access control, data encryption, controlling re-identification attacks, and anonymizing personal information [34]. Clear policies and guidelines for data collection, methods, and reasons for use and sharing are essential for developing user trust and ensuring resilience.

4.2 Algorithmic Bias and Fairness

Urban-AI also raises critical ethical issues related to algorithmic bias and discriminatory outcomes. AI systems are as biased as the data they are trained on and the algorithms running on top of them [61]. If historical data is skewed or algorithmic models are built on flawed assumptions, AI could reinforce existing inequalities and create similar ones [62]. Biased AI algorithms can result in unfair resource allocation, non-rational differences, and discriminatory policing practices through skewed public service delivery.

To mitigate these problems, data must be stripped of bias by debugging training datasets [63], using machines for bias check [64], and designing algorithms to include fairness and logical metrics that are examined periodically. Involving a broad range of stakeholders, such as impacted communities, in the AI development process can help recognize and address bias at an early stage. Documentation of bias detection assists in cutting continuous unfair practices [34].

4.3 Transparency and Interpretability

Transparency and explainability are paramount ethical issues in urban artificial intelligence. As AI systems displace humans in complex and autonomous decision-making processes [65,66], it becomes unclear why a recommendation or a choice was offered. This lack of transparency about data quality and source can weaken public confidence and impede honesty about making decisions, especially in marginalized communities.

Cities aiming to increase transparency should prioritize the development of explainable AI systems as part of open governance initiatives. This can include using interpretable AI models, designing user-friendly interfaces to explain how an AI system works, and outlining the rationale behind AI decisions [67]. The provision of updated open-access sources about AI workflow, applications, and purposes elevates the knowledge of people about its capabilities; thus, they become aware of its integrative realities and gain a sense of control. Transparency creates trust, supports effective public dialogue, and allows for responsible human involvement.

4.4 Accountability and human oversight

Accountability and human oversight are crucial in urban-AI. While increased autonomy and the impact of AI systems on decision-making together with planners in cities are generally welcomed, the method obtained in data collection is usually complex, leading to ambiguity and a lack of accountability. Thus, it is important to ensure that responsibilities are clearly articulated, and mechanisms are in place to hold both autonomous objects and their creators accountable.

Effective AI governance should include clear roles and responsibilities for the city, AI developers, and the public, as city co-creation is paramount for critical decisions to obtain optimal effectiveness. Educating parties about AI systems regulate humanistic aspirations and conceptions in parallel with the quantitative dimensions of AI [68]. This is especially essential for connecting the AI outputs of collecting and analyzing data processes with the decision-making processes. Moreover, clear procedures must be established for incidents where an AI system fails and causes harm, including investigation, identification of accountability, and remediation activities. Regular audits and external evaluations should be conducted to check the performance, fairness, and compliance of AI systems with ethical principles [34] and to also check that the data are not gathered for unethical reasons. Oversight committees to review and validate AI solutions and human roles can be assigned based on specific backgrounds and specialties.

4.5 Social Inclusion and Civic Participation

The introduction of AI into cities creates ethical issues related to social inclusion and democratic participation. Smart city initiatives often focus on technological fixes without proper engagement from the diverse communities they intend to benefit, risking the development of AI systems that do not account for marginalized or underrepresented groups [62].

To ensure that urban-AI works for all residents, cities must focus on inclusive design and a participatory approach [32, 34]. This includes comprehensively soliciting feedback and engaging all stakeholders, including low-income communities, people with disabilities, people with low access to digital services [69], and minority groups, in the development and deployment of AI. Community meetings, co-design workshops, equitable access to AI services, and citizen science projects can help involve residents in the design of AI systems that will affect their lives and cover their needs, with a reliance on building ownership and credibility [34].

Cities must also ensure that the benefits of AI are widely felt, leaving no community behind. This may entail focused investments in digital infrastructure [70], training, and support services to bring all citizens online and prepare them for the participation in and benefits of AI-enabled urban innovations.

5 Conclusion

This comprehensive review has explored the integration of AI, ML, and DL techniques in urbanism, highlighting their transformative potential in optimizing city operations and improving the quality of life for residents. The review has delved into specific ML algorithms, including RF, SVM, k-means clustering, gradient boosting machine, and time series analysis, which have demonstrated remarkable accuracy in various urban applications. Furthermore, this research has explored the application of DL algorithms, such as CNN, RNN, LSTM networks, GNN, VAE, and transformer neural networks. These advanced techniques have shown impressive performance in various urban domains. In addition, they have been shown in large-scale urban data sets, enabling more accurate predictions and optimizations.

However, the review has also emphasized the critical ethical considerations that must be addressed in the integration of AI in urbanism. With a significant portion of the world's population projected to live in urban areas by 2050, ensuring privacy, fairness, transparency, and accountability in AI-driven urban initiatives is paramount.

The paper has stressed the importance of inclusive design, participatory approaches, and targeted investments in digital infrastructure and training to ensure that the benefits of AI are widely distributed among diverse urban communities.

As the global urban population continues to grow, the responsible integration of AI in urban planning and management will be crucial for creating sustainable, resilient, and inclusive cities of the future. Further research and interdisciplinary collaboration will be essential to address the evolving landscape of urban-AI and ensure its benefits are harnessed for the well-being of all city dwellers. The potential of AI-driven techniques in building sustainable and resilient cities is immense, and this comprehensive review serves as a call for further research and development in urban-AI to address pressing urban challenges.

Acknowledgments

We would like to express our gratitude to Urban-AI 2024 organizers. We are grateful to the reviewers who contributed to the manuscript's quality improvement.

References

[1] Hu, J., Zhang, F., Qiu, B., Zhang, X., Yu, Z., Mao, Y., Wang, C., & Zhang, J. (2024). Green-gray imbalance: Rapid urbanization reduces the probability of green space exposure in early 21st century China. Science of The Total Environment, 933, 173168.

[2] Sukhova, O., & Inozemtsev, I. (2024). Regional aspects of migration and urbanization in the USSR/Russia in the 20th–21st centuries (using the example of the Penza region). BIO Web of Conferences,

[3] Balaji, S., & Babu, S. C. (2024). Approaches to population estimations and their use in food policy analysis in urban settings. In Handbook on Public Policy and Food Security (pp. 281-288). Edward Elgar Publishing.

[4] Lenhardt, A. (2024). Ending extreme poverty in an increasingly urbanised world.

[5] Herath, H. M. K. K. M. B., & Mittal, M. (2022). Adoption of artificial intelligence in smart cities: A comprehensive review. International Journal of Information Management Data Insights, 2(1), 100076. https://doi.org/10.1016/J.JJIMEI.2022.100076

[6] Ullah, Z., Al-Turjman, F., Mostarda, L., & Gagliardi, R. (2020). Applications of Artificial Intelligence and Machine learning in smart cities. Computer Communications, 154, 313–323. https://doi.org/10.1016/J.COMCOM.2020.02.069

[7] Luusua, A., Ylipulli, J., Foth, M., & Aurigi, A. (2023). Urban AI: understanding the emerging role of artificial intelligence in smart cities. AI and Society, 38(3), 1039–1044. https://doi.org/10.1007/S00146-022-01537-5/METRICS

[8] Khediri, A., Laouar, M. R., & Eom, S. B. (2021). Improving Intelligent Decision Making in Urban Planning: Using Machine Learning Algorithms. Https://Services.Igi-Global.Com/Resolvedoi/Resolve.Aspx?Doi=10.4018/IJBAN.2021070104, 8(3), 40–58. https://doi.org/10.4018/IJBAN.2021070104

[9] Jiang, F., Ma, J., Webster, C. J., Chiaradia, A. J. F., Zhou, Y., Zhao, Z., & Zhang, X. (2024). Generative urban design: A systematic review on problem formulation, design generation, and decision-making. Progress in Planning, 180, 100795. https://doi.org/10.1016/J.PROGRESS.2023.100795

[10] Kamrowska-Załuska, D., & Bibri, S. E. (2021). Impact of AI-Based Tools and Urban Big Data Analytics on the Design and Planning of Cities. Land 2021, Vol. 10, Page 1209, 10(11), 1209. https://doi.org/10.3390/LAND10111209

[11] Caprotti, F., Cugurullo, F., Cook, M., Karvonen, A., Marvin, S., McGuirk, P., & Valdez, A.-M. (2024). Why does urban Artificial Intelligence (AI) matter for urban studies? Developing research directions in urban AI research. Urban Geography, 1-12.

[12] Yigitcanlar, T., Desouza, K. C., Butler, L., & Roozkhosh, F. (2020). Contributions and risks of artificial intelligence (AI) in building smarter cities: Insights from a systematic review of the literature. Energies, 13(6), 1473.

[13] Shah, S., Rutherford, R., & Menon, S. (2020). Emerging technologies of IoT usage in global logistics. Proceedings of International Conference on Computation, Automation and Knowledge Management, ICCAKM 2020, 251–257. https://doi.org/10.1109/ICCAKM46823.2020.9051530

[14] Koohang, A., Sargent, C. S., Nord, J. H., & Paliszkiewicz, J. (2022). Internet of Things (IoT): From awareness to continued use. International Journal of Information Management, 62, 102442. https://doi.org/10.1016/J.IJINFOMGT.2021.102442

[15] Van Dinh, D., Yoon, B. N., Le, H. N., Nguyen, U. Q., Phan, K. D., & Pham, L. Di. (2020). ICT Enabling Technologies for Smart Cities. International Conference on Advanced Communication Technology, ICACT, 2020, 1180–1192. https://doi.org/10.23919/ICACT48636.2020.9061541

[16] Savastano, M., Suciu, M. C., Gorelova, I., & Stativă, G. A. (2023). How smart is mobility in smart cities? An analysis of citizens' value perceptions through ICT applications. Cities, 132, 104071. https://doi.org/10.1016/J.CITIES.2022.104071

[17] Zhao, H.-W. (2023). AI Assists Operation and Maintenance of Future Cities. Artificial Intelligence Advances, 5(1), 25–27. https://doi.org/10.30564/aia.v5i1.5780

[18] Shenavarmasouleh, F., Mohammadi, F. G., Amini, M. H., & Arabnia, H. R. (2021). Embodied AI-Driven Operation of Smart Cities: A Concise Review. Cyberphysical Smart Cities Infrastructures: Optimal Operation and Intelligent Decision Making, 29–45. https://doi.org/10.1002/9781119748342.CH3

[19] Adefemi, A., Ukpoju, E. A., Adekoya, O., Abatan, A., & Adegbite, A. O. (2023). Artificial intelligence in environmental health and public safety: A comprehensive review of USA strategies. World Journal of Advanced Research and Reviews, 20(3), 1420-1434.

[20] Bharadiya, J. (2023). Artificial Intelligence in Transportation Systems A Critical Review. American Journal of Computing and Engineering, 6(1), 34–45. https://doi.org/10.47672/ajce.1487

[21] Lv, Z., Lou, R., & Singh, A. K. (2021). AI Empowered Communication Systems for Intelligent Transportation Systems. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4579–4587. https://doi.org/10.1109/TITS.2020.3017183

[22] Iyer, L. S. (2021). AI enabled applications towards intelligent transportation. Transportation Engineering, 5, 100083. https://doi.org/10.1016/J.TRENG.2021.100083

[23] Ali, A. N. F., Sulaima, M. F., Razak, I. A. W. A., Kadir, A. F. A., & Mokhlis, H. (2023). Artificial intelligence application in demand response: advantages, issues, status, and challenges. IEEE access, 11, 16907-16922.

[24] Babu, G. R., Chintalapati, P. V., Sree, P. K., Tayaru, G., Raju, P., & Kumar, K. S. (2023). An Advanced Artificial Intelligence-Driven Smart Home Towards Ontology-Based Energy Efficiency Management System. International Conference on Innovations in Data Analytics,

[25] Jacobs, G., Van Houdt, F., & coons, g. (2024). Studying Surveillance AI-cologies in Public Safety: How AI Is in the World and the World in AI. Surveillance & Society, 22(2), 160-178.

[26] Anwar, M. R., & Sakti, L. D. (2024). Integrating Artificial Intelligence and Environmental Science for Sustainable Urban Planning. IAIC Transactions on Sustainable Digital Innovation (ITSDI), 5(2), 179-191.

[27] Chakraborty, N., Mishra, Y., Bhattacharya, R., & Bhattacharya, B. (2023). Artificial Intelligence: The road ahead for the accessibility of persons with Disability. Materials Today: Proceedings, 80, 3757–3761. https://doi.org/10.1016/J.MATPR.2021.07.374

[28] Awogbemi, O., Kallon, D. V. Von, & Kumar, K. S. (2024). Contributions of artificial intelligence and digitization in achieving clean and affordable energy. Intelligent Systems with Applications, 22, 200389. https://doi.org/10.1016/J.ISWA.2024.200389

[29] Philips, A., Bharadwaj Thuraka, A. ;, Blossom, S., & Okpeseyi, A. (2024). Achieving Housing Affordability in the U.S. through Sustained Use of AI and Robotic Process Automation for Prefabricated Modular Construction. African Journal of Advances in Science and Technology Research, 15(1), 122–134. https://doi.org/10.62154/53T99N63

[30] Cachat-Rosset, G., & Klarsfeld, A. (2023). Diversity, Equity, and Inclusion in Artificial Intelligence: An Evaluation of Guidelines. Applied Artificial Intelligence, 37(1). https://doi.org/10.1080/08839514.2023.2176618

[31] Fosch-Villaronga, E., & Poulsen, A. (2022). Diversity and Inclusion in Artificial Intelligence. 109–134. https://doi.org/10.1007/978-94-6265-523-2_6

[32] Huriye, A. Z. (2023). The Ethics of Artificial Intelligence: Examining the Ethical Considerations Surrounding the Development and Use of AI. American Journal of Technology , 2(1), 37–44. https://doi.org/10.58425/ajt.v2i1.142

[33] Hastuti, R., & Syafruddin. (2023). Ethical Considerations in the Age of Artificial Intelligence: Balancing Innovation and Social Values. West Science Social and Humanities Studies , 1(02), 76–87. https://doi.org/10.58812/WSSHS.V1I02.191

[34] Sanchez, T. W., Brenman, M., & Ye, X. (2024). The Ethical Concerns of Artificial Intelligence in Urban Planning. Journal of the American Planning Association, 0(0), 1–14. https://doi.org/10.1080/01944363.2024.2355305

[35] Kitchin, R. (2016). The ethics of smart cities and urban science. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2083). https://doi.org/10.1098/RSTA.2016.0115

[36] Pastor-Escuredo, D., Treleaven, P., & Vinuesa, R. (2022). An Ethical Framework for Artificial Intelligence and Sustainable Cities. AI 2022, Vol. 3, Pages 961-974, 3(4), 961–974. https://doi.org/10.3390/AI3040057

[37] Chowdhury, M. S. (2024). Comparison of accuracy and reliability of random forest, support vector machine, artificial neural network and maximum likelihood method in land use/cover classification of urban setting. Environmental Challenges, 14, 100800.

[38] Sun, S., Yan, H., & Lang, Z. (2024). A study on traffic congestion prediction based on random forest model. *Highlights in Science, Engineering and Technology*, *101*, 738-749.

[39] Ramachandra, T. V., Mondal, T., & Setturu, B. (2023). Relative performance evaluation of machine learning algorithms for land use classification using multispectral moderate resolution data. *SN Applied Sciences*, *5*(10), 274.

[40] Wang, H., Cao, J., Huang, R., Liu, L., & Wang, H. (2022, March). Analyze and Predict Traffic Flow by Using SVM Model. In *ECITech 2022; The 2022 International Conference on Electrical, Control and Information Technology* (pp. 1-5). VDE.

[41] Toan, T. D., & Truong, V. H. (2021). Support vector machine for short-term traffic flow prediction and improvement of its model training using nearest neighbor approach. *Transportation research record*, *2675*(4), 362-373.

[42]Kranjčić, N., Medak, D., Župan, R., & Rezo, M. (2019). Support vector machine accuracy assessment for extracting green urban areas in towns. *Remote Sensing*, *11*(6), 655.

[43] Chowdhury, M. S. (2024). Comparison of accuracy and reliability of random forest, support vector machine, artificial neural network and maximum likelihood method in land use/cover classification of urban setting. *Environmental Challenges*, *14*, 100800.

[44] Kilani, B. H. (2023). K-Means Clustering algorithms in Urban studies: A Review of Unsupervised Machine Learning techniques.

[45] Li, Y., Zhou, X., Gu, J., Guo, K., & Deng, W. (2022). A novel K-means clustering method for locating urban hotspots based on hybrid heuristic initialization. *Applied Sciences*, *12*(16), 8047.

[46] Yuan, H., Wang, M., Zhang, D., Ikram, R. M. A., Su, J., Zhou, S., ... & Zhang, Q. (2024). Data-driven urban configuration optimization: An XGBoost-based approach for mitigating flood susceptibility and enhancing economic contribution. *Ecological Indicators*, *166*, 112247.

[47] Salim, K., Hebri, R. S. A., & Besma, S. (2022). Classification predictive maintenance using XGboost with genetic algorithm. *Revue d'Intelligence Artificielle*, *36*(6), 833.

[48] Mirindi, D., Sanders, T. N., Oshineye, O., & Hunter, J. (2024). Integration of Artificial Intelligence and Smart Technologies in Offsite Construction: A Comprehensive Review. Transforming Construction with Off-site Methods and Technologies.

[49] Tran, H.-D., Bak, S., Xiang, W., & Johnson, T. T. (2020). Verification of deep convolutional neural networks using imagestars. International conference on computer aided verification,

[50] Boston, T., Van Dijk, A., Larraondo, P. R., & Thackway, R. (2022). Comparing CNNs and random forests for Landsat image segmentation trained on a large proxy land cover dataset. Remote Sensing, 14(14), 3396.

[51] Batbaatar, E., Park, H. W., Li, D., Li, M., & Ryu, K. H. (2018). DeepEnergy: Prediction of appliances energy with long-short term memory recurrent neural network. Asian Conference on Intelligent Information and Database Systems,

[52] Rao, K. S., Devi, G. L., & Ramesh, N. (2019). Air quality prediction in Visakhapatnam with LSTM based recurrent neural networks. International Journal of Intelligent Systems and Applications, 11(2), 18-24.

[53] Tian, Y., & Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. 2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity),

[54] Yan, C., Wang, S., Yang, J., Xu, T., & Huang, J. (2020). Re-balancing variational autoencoder loss for molecule sequence generation. Proceedings of the 11th ACM international conference on bioinformatics, computational biology and health informatics,

[55] Liang, F., Qian, C., Yu, W., Griffith, D., & Golmie, N. (2022). Survey of graph neural networks and applications. Wireless Communications and Mobile Computing, 2022(1), 9261537.

[56] Vaswani, A. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.

[57] Chimalamarri, S., Sitaram, D., Mali, R., Johnson, A., & Adeab, K. A. (2020, December). Improving Transformer based Neural Machine Translation with Source-side Morpho-linguistic Features. In *2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)* (pp. 1-5). IEEE.

[58] Kashef, M., Visvizi, A., & Troisi, O. (2021). Smart city as a smart service system: Human-computer interaction and smart city surveillance systems. Computers in Human Behavior, 124, 106923. https://doi.org/10.1016/J.CHB.2021.106923

[59] Braun, T., Fung, B. C. M., Iqbal, F., & Shah, B. (2018). Security and privacy challenges in smart cities. Sustainable Cities and Society, 39, 499–507. https://doi.org/10.1016/J.SCS.2018.02.039

[60] Jobin, A., Ienca, M., & Vayena, E. (2019). AI: the global landscape of ethics guidelines. Nature Machine Intelligence, 1(9), 389–399. https://arxiv.org/pdf/1906.11668

[61] Xia, L., Semirumi, D. T., & Rezaei, R. (2023). A thorough examination of smart city applications: Exploring challenges and solutions throughout the life cycle with emphasis on safeguarding citizen privacy. Sustainable Cities and Society, 98, 104771. https://doi.org/10.1016/J.SCS.2023.104771

[62] Du, J., Ye, X., Jankowski, P., Sanchez, T. W., & Mai, G. (2024). Artificial intelligence enabled participatory planning: a review. International Journal of Urban Sciences, 28(2), 183–210. https://doi.org/10.1080/12265934.2023.2262427

[62] Veale, M., & Binns, R. (2017). Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. Big Data and Society, 4(2), 1–17. https://doi.org/10.1177/2053951717743530

[63] Corbett-Davies, S., & Goel, S. (2018). The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. ArXiv.Org.

[64] Shahbazi, N., Lin, Y., Asudeh, A., & Jagadish, H. V. (2023). Representation Bias in Data: A Survey on Identification and

Resolution Techniques. ACM Computing Surveys, 55(13 s). https://doi.org/10.1145/3588433
[65] Araujo, T., Helberger, N., Kruikemeier, S., & de Vreese, C. H. (2020). In AI we trust? Perceptions about automated decision-making by artificial intelligence. AI and Society, 35(3), 611–623. https://doi.org/10.1007/S00146-019-00931-W/METRICS
[66] Schmidt, P., Biessmann, F., & Teubner, T. (2020). Transparency and trust in artificial intelligence systems. Journal of Decision Systems, 29(4), 260–278. https://doi.org/10.1080/12460125.2020.1819094
[67] Hersperger, A. M., & Fertner, C. (2021). Digital plans and plan data in planning support science. Environment and Planning B: Urban Analytics and City Science, 48(2),212–215. https://doi.org/10.1177/2399808320983002/ASSET/2399808320983002.FP.PNG_V03
[68] Sanchez, T. W., Shumway, H., Gordner, T., & Lim, T. (2023). The prospects of artificial intelligence in urban planning. International Journal of Urban Sciences, 27(2), 179–194. https://doi.org/10.1080/12265934.2022.2102538
[69] Afzalan, N., & Muller, B. (2018). Online Participatory Technologies: Opportunities and Challenges for Enriching Participatory Planning. Journal of the American Planning Association, 84(2), 162–177. https://doi.org/10.1080/01944363.2018.1434010
[70] Malik, J., & Zhao, X. (2023). Does the new digital infrastructure improve total factor productivity? Bulletin of Economic Research, 75(4), 895–916. https://doi.org/10.1111/BOER.12388

Generative-AI based Map Representation and Localization

Alireza Akhavi Zadegan*
Amnir Hadachi*
alireza.akhavi.zadegan@ut.ee
hadachi@ut.ee
ITS Lab, Institute of Computer Science, University of Tartu
Tartu, Estonia

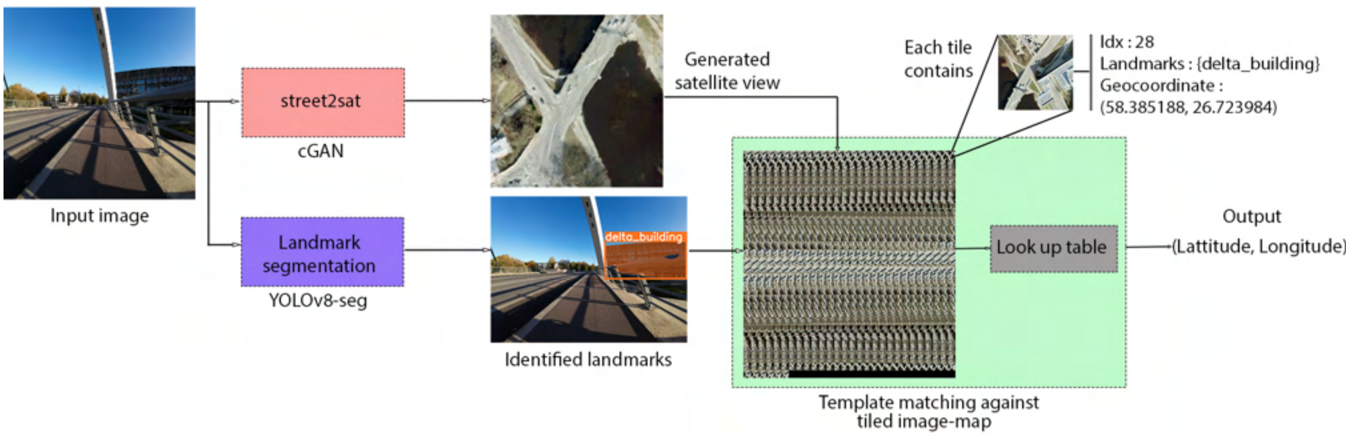


Figure 1. An illustration of the image based ground-to-satellite geolocalization problem, and our proposed framework

Abstract

In the domain of Image-Based Localization (IBL), the precise integration of street-level and satellite perspectives plays a pivotal role, particularly in dynamic urban environments. This research introduces a novel generative AI framework that systematically transforms street-level imagery into corresponding satellite views, thereby bridging the observational gap between ground and overhead perspectives. Such capability alongside known urban landmarks holds promise for constructing contextual reference databases and improving image-based localization in ever-changing urban environments. Central to our methodology is the creation of a contextual tiled image-map, offering a novel perspective on urban mapping. We systematically organize a specific urban area into a structured sequence of satellite-simulated image tiles derived from ground-level data using the street2sat generative model. Each tile is indexed by its direction of travel, geographically tagged, and annotated with visible landmarks identified through a custom landmark segmentation model, creating a comprehensive contextual database. The localization process initiates with the submission of street-level image to a landmark segmentation model, which identifies recognizable landmarks within the image. These landmarks refine the subsequent search process by pinpointing tile locations in the image-map associated with similar landmarks, narrowing down the search scope to a set of valid indexes. Following the initial step, the image query is processed by our street2sat generative model, which transforms the street-level image into a corresponding satellite view. This view is then matched against a collection of pre-validated tiles to find the closest match. Finally, the input image is given a matching latitude and longitude by computing the tile number, which is linked to a particular geocoordinate.

CCS Concepts: **• Computing methodologies → Visual content-based indexing and retrieval**.

Keywords: Generative-AI, Mapping, Localization

UrbanAI'24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1156-5/24/10
https://doi.org/10.1145/3681780.3697276

ACM Reference Format:
Alireza Akhavi Zadegan and Amnir Hadachi. 2024. Generative-AI based Map Representation and Localization. In *2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI (UrbanAI'24), October 29-November 1, 2024, Atlanta, GA, USA*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3681780.3697276

1 Introduction

In the interdisciplinary nexus of computer vision, robotics, and navigation, IBL has emerged as a cornerstone technology, supporting a wide array of applications from augmented reality and advanced navigation systems to the autonomy of robots [4, 18]. IBL is pivotal for determining the precise position and orientation of the camera within a predefined space, employing either individual or sequences of images to enable precise and context-aware navigation within complex environments. The quest for visual localization algorithms, essential for the spatial understanding needed in robotics and autonomous vehicles, presents a unique blend of challenges, encompassing complex image matching and the analysis of comprehensive 3D data like point clouds and visual descriptors. The creation of detailed 3D maps through technologies such as LiDAR and photogrammetry, while valuable, faces notable challenges due to their significant costs, the necessity for continual updates, and the extensive storage requirements often demanding cloud-based solutions that limit on-device localization and scalability [5, 12, 23]. This scenario has prompted a paradigm shift from traditional visual localization techniques, which typically involved aligning a query image with a sparse Structure-from-Motion (SfM) model of the scene, towards learning-based methodologies that offer enhanced scalability and adaptability [6, 16, 24, 26]. However, even state-of-the-art tools like HLoc encounter difficulties with scene variations and the 2D-3D matching phase, primarily due to their dependence on multiple 2D-2D image matchings [17]. Simultaneously, a keen interest has emerged in cross-view ground-to-satellite localization as a more compact yet costly alternative to expansive 3D maps. While this approach reduces spatial data to a more manageable size, it often comes with high financial and storage burdens, particularly at finer resolutions [7, 11, 20], and is prone to less accurate orientation results [25]. Meanwhile, other methodologies have turned to 3D metric sensors such as LiDAR [3, 13] and radar [1, 21] for their precise template matching capabilities, albeit at considerable expense.

In the context of IBL, diverse map representations each carry distinct advantages and challenges. High-resolution 3D maps, derived from ground images and often employing SfM techniques for 2D-3D correspondences, are invaluable for accurate camera pose estimation [15]. Though these maps are rich in localization information, their creation and maintenance are resource-intensive. Conversely, 2D satellite images provide broader coverage and unique aerial perspectives but are limited by their level of detail and struggle to capture the dynamic urban environment [11, 26]. Planimetric maps from sources like OpenStreetMap offer essential layout details and are widely accessible, yet lack crucial information such as elevation and finer environmental details, which are vital for tackling urban localization complexities [14, 19].

Our methodology leverages generative AI alongside custom landmark segmentation model to bridge the gap between traditional mapping and localization techniques, by transforming street-level images into satellite-like views, we generate comprehensive top-down perspectives essential for accurate localization, particularly in complex urban environments. This aerial perspective offers a broader field of vision, revealing spatial relationships and key landmarks obscured at ground level. By combining these detailed, bird's-eye views with ground-level information, our approach enhances feature matching accuracy and overall localization performance. This innovative process not only improves map representation but also offers significant advantages in terms of cost-efficiency, scalability, and adaptability to dynamic urban settings. Additionally, generative AI's capability to integrate diverse forms of data, including topographical and infrastructural information [9], results in the creation of more detailed and useful maps, showcasing its transformative potential in navigating and understanding complex urban settings.

2 Framework Overview

Our framework is designed for specific urban setting, integrating a blend of advanced deep learning models with image template matching algorithms to enhance mapping and localization capabilities. The process commences with the collection of georeferenced images from two primary perspectives that thoroughly encompass the target area. This method emulates the navigational patterns of delivery robots and micro-mobility solutions, offering a comprehensive view of the urban terrain. Following data collection, we developed two specialized models. The first is a custom landmark segmentation model, designed to identify specific urban landmarks such as buildings, statues, and bridges. The second is a generative model, named street2sat, which transforms the collected street-level images into corresponding satellite views, effectively bridging the gap between ground and satellite perspectives. Subsequently, we integrate all the data into a single, compact reference database known as the contextual tiled image-map. This database organizes the region of interest into a sequence of downsampled, simulated satellite tiles, each aligned according to their path trajectory and timestamps. Each tile is enriched with essential information, including key landmarks, geocoordinate points, and positional indices, creating a comprehensive and detailed reference source.

Transitioning from mapping to localization, the process begins with the input of a street-level image into our landmark segmentation model, which detects distinct landmarks. These landmarks are crucial for refining the search within the tiled image-map, directing our template matching algorithm toward tiles displaying similar landmarks and thereby

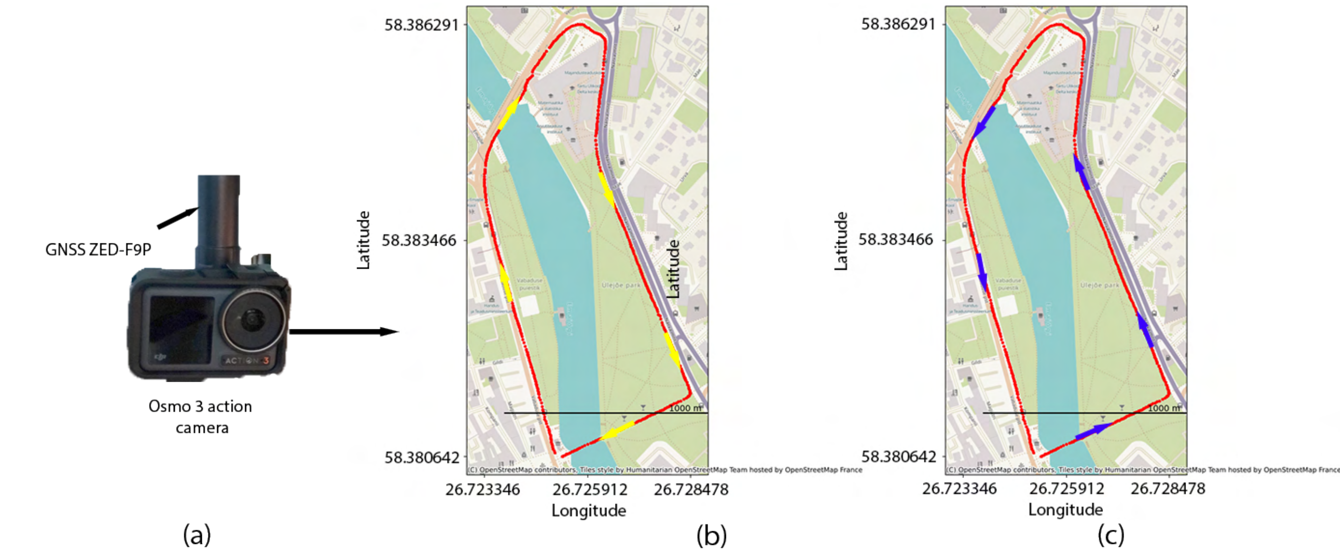


Figure 2. Visual representation for data collection procedure: (a) sensor capturing setup, (b) path taken for data collection in clockwise direction, indicated by yellow arrows, (c) path taken for data collection in a counter-clockwise direction, marked by blue arrows.

narrowing the search to a targeted subset of indexes. Subsequently, the image is processed by street2sat, which converts the street-level image into its satellite equivalent. The most suitable match is then determined by comparing this satellite view with a collection of pre-validated tiles in the image-map. Once the best match is identified, the system assigns latitude and longitude to the given image by pinpointing the specific tile number associated with unique geographical coordinates. Figure 1 provides a visual representation of the procedure.

2.1 Data Collection

The data collection for our study utilized Osmo 3 action camera paired with a ZED-F9P GNSS module, as illustrated in Figure 2 (a). Using this configuration we georeferenced images at intervals averaging 3.89 meters between each shot. We conducted this process through two exhaustive sweeps of the region of interest: initially in a clockwise direction and subsequently in a counter-clockwise manner, as shown in Figures 2 (b) and 2 (c). Mirroring the typical navigation patterns of delivery robots and micro-mobility devices, sought to acquire dual perspectives from identical locations, ensuring a wide array of visual cues encountered on sidewalks and paths were documented. At each capture point, we collected three types of data: the RGB image, the geocoordinates, and the timestamps. Finally, we synchronized the RGB images with their corresponding geocoordinates using the timestamps to ensure temporal alignment.

2.2 Landmark Segmentation

Custom landmark segmentation in IBL primarily serves to enhance the robustness and adaptability of localization systems in diverse and dynamic environments. By accurately identifying and segmenting distinct landmarks and features within urban landscapes, the system can create more reliable and context-aware matches between the query image and the database. This approach not only increases precision in pinpointing locations but also improves the system's ability to function effectively under varied conditions, including occlusions, lighting changes, and seasonal variations.

For the annotation of our dataset, we utilized the Segment Anything Model (SAM) [10], focusing on 27 distinct landmarks relevant to our study area. The task involved manually identifying and segmenting these landmarks in approximately 800 street-level images, significantly simplifying the typically demanding labeling process. The utilization of SAM's advanced zero-shot learning capabilities enabled us to expedite our data preparation, playing a pivotal role in reducing complexity and improving the accuracy of localization in urban environments. Following the labeling process, we used the generated annotations to train an instant segmentation model based on the YOLOv8 architecture [22]. The model was set to train for 100 epochs, incorporating an early stopping mechanism with a patience parameter set to 50 epochs to prevent overfitting. The training was conducted with a batch size of 16. Notably, the training was driven by a Stochastic Gradient Descent (SGD) optimizer and was conducted without relying on a pretrained model, ensuring our

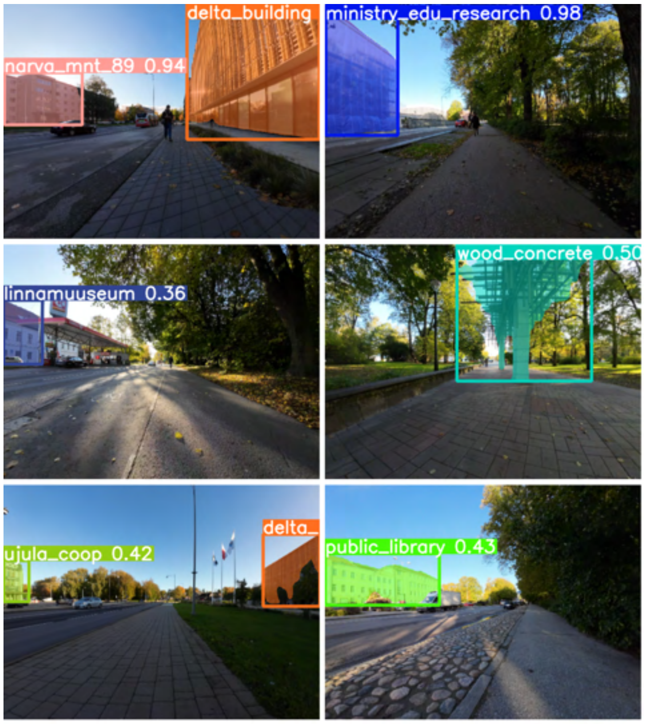


Figure 3. Examples of successfully identified landmarks using our custom landmark segmentation model on street-level images in Tartu, Estonia.

model was uniquely tailored to our dataset. The effectiveness of our segmentation model is visually represented in Figure 3.

To assess the effectiveness of our semantic segmentation method, we evaluated its precision and mean Average Precision (mAP) for both bounding boxes and masks over several epochs. These metrics serve as essential indicators of the model's proficiency in identifying and categorizing landmarks within urban infrastructure analysis. Our evaluation specifically concentrated on Precision (B) and mAP50 (B) for bounding boxes, along with Precision (M) and mAP50 (M) for masks. The analysis of results from the initial, median, and final epochs reveals a significant enhancement in the model's capacity for precise segmentation and landmark recognition. This improvement is detailed in Table 1, providing an overview of the segmentation quality metrics across several epochs.

2.3 street2sat Generative Model

To configure the street2sat generative model, which is designed to transform street-level images into their corresponding satellite views, we employed the pix2pix cGAN framework [8]. This setup involves pairing each street-level image with its corresponding satellite view, anchored by geographic

Table 1. Overview of our custom segmentation performance metrics across several epochs.

Metric	Initial	Median	Final
Precision (B)	0.69	0.78	0.76
mAP50 (B)	0.43	0.76	0.79
Precision (M)	0.69	0.78	0.73
mAP50 (M)	0.42	0.76	0.75

coordinates obtained from a GNSS module. These coordinates delineate the center of a designated square region of interest, mathematically defined as:

$$Square; Area = [(lon - d, lat - d), (lon + d, lat + d)] \quad (1)$$

To optimize the model's performance during the training phase, we implemented specific configurations. The model underwent training with a batch size of 1 and a learning rate (lr) set at 0.0002, adhering to best practices for achieving stable GAN training outcomes. The initial training phase spanned 500 epochs, after which the model's parameters were further refined through an additional 500 epochs in a decay phase to fine-tune performance. The adjustment of the learning rate over these epochs follows a linear decay function:

$$lr = lr_{initial} - \frac{epoch \times lr_{initial}}{n_{epochs} + n_{epochs_decay}} \quad (2)$$

In this equation, $lr_{initial}$ represents the starting learning rate, $epoch$ the current epoch count, n_{epochs} the total epochs before decay begins, and n_{epochs_decay} the duration of the decay phase. Consistency across the dataset was maintained by resizing each image to a square dimension of 256×256 pixels, streamlining the training process. The transformation of street-level images to satellite views serves several critical purposes. Firstly, it provides a comprehensive top-down perspective of a location, essential for the identification and mapping of key landmarks and geographic features vital for accurate localization, especially in urban environments where obstacles and dynamic changes pose challenges to conventional localization approaches. Secondly, the aerial perspective broadens the field of vision, unveiling the spatial relationships and arrangements among objects and features that ground-level views might obscure. This broadened view is crucial for enhancing feature matching accuracy, as it compiles a diverse and detailed dataset that mirrors the complexity of real-world settings. By capturing both the detailed and overarching aspects of a location, our system enhances its navigational and localization precision, leveraging various visual cues from different viewpoints to navigate and understand complex environments effectively. Figure 4 (c) demonstrates the output of our street2sat generative model alongside its actual satellite view (Figure 4 (b)) and its corresponding street-level image (Figure 4 (a)).

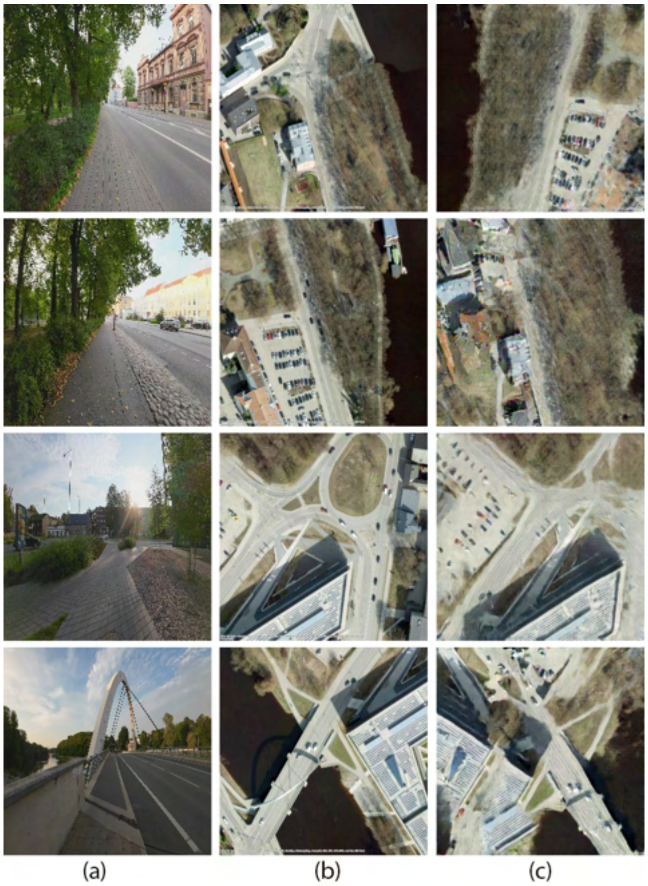


Figure 4. (a) street-level image, (b) actual satellite view from same location extracted from OpenStreetMap, (c) our street2sat generated satellite view from the street-level image.

2.4 Contextual Tiled Image-Map Database

The concept of the contextual tiled image-map represents an innovative approach to mapping an entire region through an ordered sequence of satellite view images, each rich with contextual data. Each tile in the image-map carries essential information: its position index relative to other tiles, its specific geolocation, and the landmarks within its field of view, as depicted in Figure 5. Within our database, the ordered arrangement of tiles, combined with the distinct landmarks identified on each, offers a rough yet extremely useful view of a vehicle's trajectory and orientation. This structured compilation effectively creates a detailed mosaic that mirrors the real-world environment, capturing the essence of the urban landscape through which the vehicle navigates.

Our methodology has the potential to enhance route visualization and also significantly improve navigation predictability by depicting potential paths and adjustments a vehicle might make in an urban landscape. Moreover, this approach allows for representing a vast region as a single, detailed image, effectively condensing a broad area into a manageable format. This compact representation is particularly advantageous for memory constraints in autonomous delivery robots, as it reduces the data load without sacrificing detail or navigational utility. Building on this approach, the entire training dataset undergoes a comprehensive synthesis, utilizing two core models: the street2sat for satellite view generation and the landmark segmentation model for precise feature identification. For every street-level image in the training dataset, the street2sat model generates a corresponding (100×100) pixel satellite image, creating pivotal reference points within our localization map. Simultaneously, the landmark segmentation model processes the same street-level images, extracting specific landmarks associated with their locations. Next, we need to calculate the grid's dimensions, the aim is to organize the images in an approximately square grid to optimize space and maintain aspect ratios. The number of columns *cols* and rows *rows* is calculated using the formulas:

$$cols = \left\lfloor \sqrt{num\_images} \right\rfloor, \quad rows = \left\lceil \frac{num\_images}{cols} \right\rceil \tag{3}$$

The canvas on which the images will be tiled is then determined by the dimensions of this grid. The width and height of the canvas are directly proportional to the number of columns and rows and the dimensions of each tile, described by:

$$canvas\_width = cols \times tile\_width \tag{4}$$

$$canvas\_height = rows \times tile\_height \tag{5}$$

As each image undergoes processing, it is resized to fit the predetermined tile dimensions and then accurately placed onto the canvas. The positioning formulae ensure that each image is allocated to its precise location on the grid, based on its sequence and the overall structure of the tile arrangement:

$$x_{pos} = (i \mod cols) \times tile\_width \tag{6}$$

$$y_{pos} = \left\lfloor \frac{i}{cols} \right\rfloor \times tile\_height \tag{7}$$

2.5 Localization by Template Matching

In order to carry out localization in a contextual tiled image-map, we begin by feeding an image query into two essential procedures. Initially, the image is introduced to a landmark segmentation model, which discerns any recognizable landmarks within the image. The landmarks identified are crucial; they are utilized to refine the search process of the template matching algorithm by focusing on tile locations within the image-map associated with similar landmarks. This refinement is represented as a narrowed set of valid indexes V derived from landmark data. Next, the street-level input image is transformed into its corresponding satellite view using

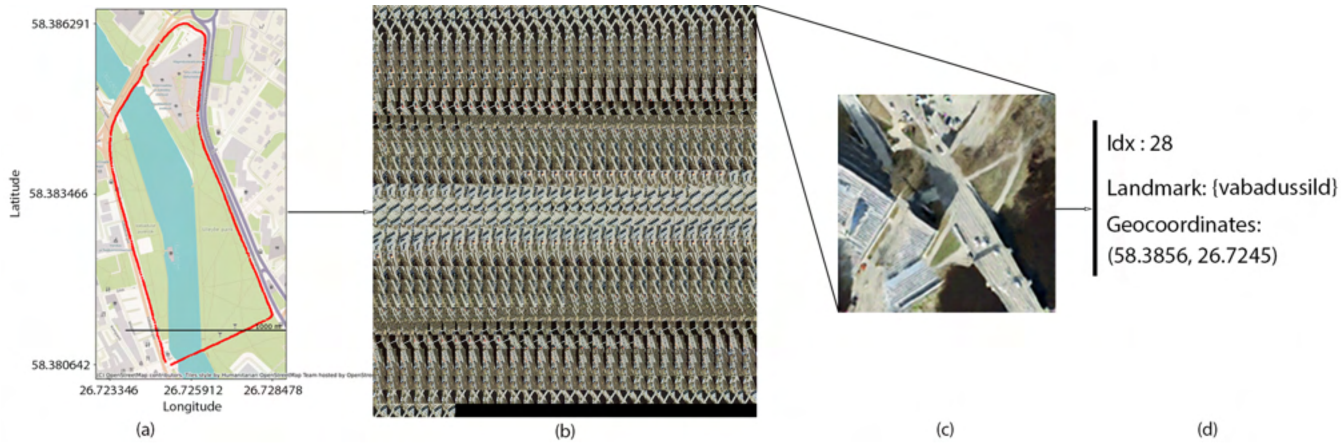


Figure 5. (a) An overview of the area under study using OpenStreetMap, (b) the entire study region presented as a composite synthesized image using the street2sat model, (c) a sample image-tile, (c) details contained in every image-tile, including its position with relation to other tiles, landmarks visible from the street-level perspective, and the geocoordinates of the location where the image was captured.

the street2sat model, denoted as S. The generated satellite image is then fed to the template matching algorithm which at its core employs a Normalized Cross-Correlation (NCC) to systematically compare S against each filtered tiles of the tiled image-map from the previous stage . The correlation coefficient for each tile is computed using the equation:

$$R(x, y) = \frac{\sum_{x',y'} [S(x', y') \cdot I(x + x', y + y')]}{\sqrt{\sum_{x',y'} S(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}} \quad (8)$$

Here, I represents the reference image, and x', y' are coordinates relative to the segment. The search for the best match is concentrated over those regions indicated by V, the set of tile locations linked to the recognized landmarks. Once the algorithm identifies the tile with the highest correlation coefficient as the best match, it determines its precise geographical location by computing the tile number using Equation 9. Subsequently, it cross-references a lookup table to retrieve the corresponding latitude and longitude associated with that positional index.

$$TileNumber = \left(\frac{y}{step\_size}\right) \cdot row\_steps + \left(\frac{x}{step\_size}\right) + 1 \quad (9)$$

Leveraging both the landmark data for spatial context and the generated satellite view for visual comparison, our approach ensures a targeted and efficient search within the template matching process.

3 Experiment and Results

To evaluate the performance of our localization methodology, we utilized a small sample from the DELTA dataset [2], which provides a comprehensive set of street-level images each paired with precise geolocations, serving as our benchmark. Initially, each image was processed through our custom landmark segmentation model, followed by the street2sat generative model. This sequential processing allowed us to refine the input for geolocation prediction. We calculated geographic distances between predicted and actual geolocations using the Haversine formula, forming the basis of our error analysis. These measurements, along with additional statistical metrics, were used to assess the accuracy of our geolocation predictions, as detailed in Tables 2 and 3.

3.1 Haversine Distance Calculation:

The Haversine formula is used to calculate the great-circle distance between two points on the earth's surface, given their latitude and longitude in degrees. This provides us with the actual distance errors between the true locations and the estimated locations from our model. The formula used for each pair of points is:

$$\begin{aligned} d &= 2 \times 6371 \times \arcsin\left(\sqrt{a}\right) \\ a &= \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}_1) \times \cos(\text{lat}_2) \times \sin^2\left(\frac{\Delta\text{lon}}{2}\right) \end{aligned} \quad (10)$$

Where Δlat and Δlon are the differences in latitude and longitude, respectively, and 6371 is the earth's radius in kilometers.

3.2 Mean Absolute Error (MAE) and Root Mean Square Error (RMSE):

These metrics give a sense of the average error and the square root of the average of squared differences between true and

Table 2. Haversine Distance Calculation Results

Metric	Value (km)
Mean Error	0.1175
Median Error	0.0739
Standard Deviation of Error	0.1219

estimated values, respectively. MAE provides a linear score that averages the absolute differences between the predicted values and actual values, giving an idea of how wrong the predictions were, without considering the direction. The formulas for these calculations are:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{12}$$

Where y_i and x_i are the true and estimated geocoordinate values, respectively.

Table 3. MAE and RMSE Results

Metric	Latitude	Longitude
MAE	0.00093	0.00070
RMSE	0.00141	0.00107

The evaluation of our localization algorithm, conducted on a modest sample size, presents promising results, demonstrating a generally reliable performance. Analysis using Haversine distance calculations shows that the algorithm's predictions typically closely align with actual locations, with a significant portion being near the true positions. The observed standard deviation indicates a reasonable range of accuracy across various predictions, highlighting some variability in precision. Despite the small sample size used for testing, these findings underscore the algorithm's potential to offer accurate and dependable location estimations in a particular urban setting.

4 Failure Analysis

In our localization prediction system, we identified two primary root causes of inaccuracies. The first issue is unsuccessful landmark classification, which arises from several factors. Some locations may lack identifiable landmarks (Figure 6 (a)), leaving our system without essential spatial context for accurate localization. In other instances, even if landmarks exist, they might be obscured by trees or urban elements (Figures 6 (b) and 6 (c)), preventing the system from recognizing them. Additionally, the system might fail to recognize the

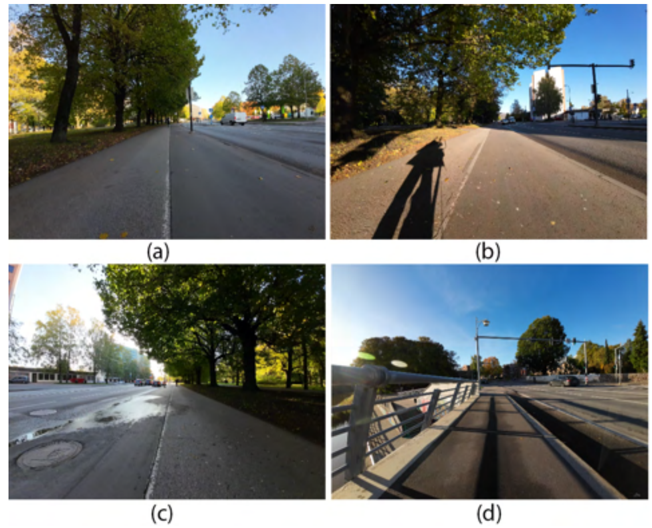


Figure 6. Examples of system failure analysis:(a) Absence of visible identifiable landmarks, (b) landmarks obscured by trees, (c) interference from urban elements, (d) reduced visibility caused by sun glare, and limitations due to insufficient model training.

landmarks due to various limitations, such as poor visibility or insufficient model training (Figure 6 (d)).

The second major issue is the inaccurate generation of satellite views by the street2sat model (Figure 7). If the model generates an imprecise representation of the urban landscape, it adversely affects the subsequent template matching process, leading to errors in localization. These fundamental challenges underscore the importance of addressing both the landmark classification accuracy and the quality of generated satellite views to enhance the overall precision of our geolocation prediction system.

5 Conlucion and future scope of research

In this research, we embarked on an innovative exploration of IBL leveraging generative AI in tandem with image template matching and landmark segmentation, specifically aimed at enhancing urban navigation for micro-mobility vehicles and delivery robots. Our method systematically creates a contextual tiled image-map by organizing a specified urban area into a structured sequence of satellite-generated image tiles, each derived from street-level imagery through the street2sat generative model. These tiles are indexed by their direction of travel, geographically tagged, and annotated with visible landmarks from the street view, culminating in a comprehensive contextual database. Our findings demonstrate the potential of generative AI to bridge the gap between ground-level observations and overhead perspectives, offering a significant promise for constructing contextual reference databases and improving image-based localization in dynamically changing urban environments.

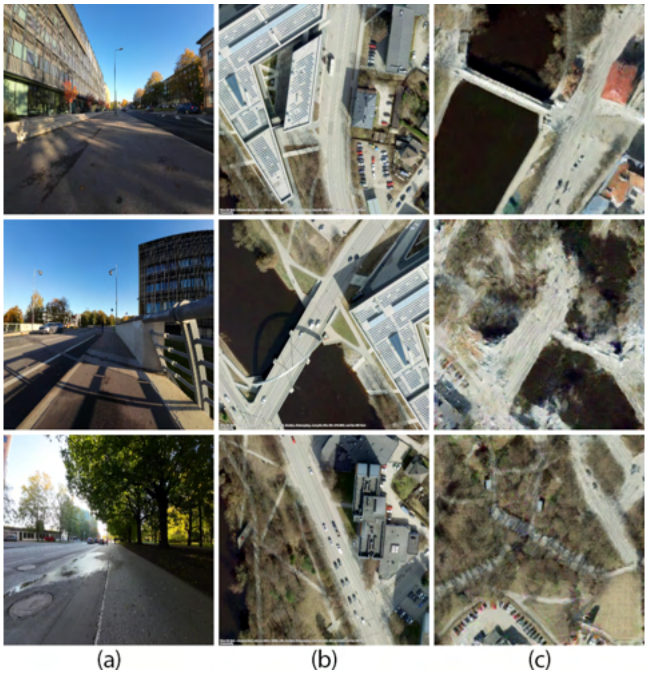


Figure 7. street2sat model failing to properly generate satellite views from the street-level view. (a) street-level view from DELTA dataset, (b) the actual satellite view from the street-level view, (c) the satellite view generated by the street2sat model.

The evaluation of our system, particularly through the use of the DELTA dataset, revealed a moderate level of accuracy, indicating room for enhancement in achieving high precision in real-world applications. The current approach to IBL, while indicative of potential, presents multiple pathways for enhancement and further exploration. Future research could prioritize the expansion of the landmark segmentation model to incorporate a more extensive array of landmarks. This would likely improve the granularity and specificity of the localization process, allowing for a more refined and detailed understanding of urban scenes. Additionally, the inclusion of more sophisticated scene understanding neural network models could provide deeper insights into complex visual data, contributing to a more robust and nuanced localization mechanism.

Improvements to the cGAN model are also warranted, particularly in its ability to handle diverse and dynamic urban scenarios. Advancements in this area could lead to better generalization capabilities, enabling the model to produce more accurate aerial representations from street-level imagery under varying conditions. Refining the model to reduce its computational demands could also make the system more viable for real-time applications on a range of devices.

Moreover, enhancing the system's capability to update its database in response to rapid environmental changes remains a critical area of development. This would ensure that the localization remains accurate over time, even as urban landscapes evolve. Finally, as the model becomes more intricate, the computational efficiency will be vital to maintain, ensuring that improvements in accuracy do not come at an unsustainable cost in terms of processing power. Addressing these areas could greatly increase the applicability and reliability of IBL technologies in real-world settings.

References

[1] Daniel Adolfsson, Martin Magnusson, Anas Alhashimi, Achim J Lilienthal, and Henrik Andreasson. 2022. Lidar-level localization with radar? the cfear approach to accurate, fast, and robust large-scale radar odometry in diverse environments. *IEEE Transactions on robotics* 39, 2 (2022), 1476–1495.

[2] Alireza Akhavi Zadegan, Damien Vivet, and Amnir Hadachi. 2024. DELTA: Integrating Multimodal Sensing with Micromobility for Enhanced Sidewalk and Pedestrian Route Understanding. *Sensors* 24, 12 (2024), 3863.

[3] Xieyuanli Chen, Ignacio Vizzo, Thomas Läbe, Jens Behley, and Cyrill Stachniss. 2021. Range image-based LiDAR localization for autonomous vehicles. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5802–5808.

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 224–236.

[5] Simon Fuhrmann, Fabian Langguth, and Michael Goesele. 2014. Mve-a multi-view reconstruction environment. *GCH* 3 (2014), 4.

[6] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. 2020. Self-supervising fine-grained region similarities for large-scale image localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 369–386.

[7] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. 2018. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7258–7267.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[9] Yuhao Kang, Song Gao, and Robert Roth. 2022. A review and synthesis of recent GeoAI research for cartography: Methods, applications, and ethics. In *Proceedings of AutoCarto*. 2–4.

[10] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

[11] Liu Liu and Hongdong Li. 2019. Lending orientation to neural networks for cross-view geo-localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5624–5633.

[12] Liu Liu, Hongdong Li, and Yuchao Dai. 2017. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *Proceedings of the IEEE International Conference on Computer Vision*. 2372–2381.

[13] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. 2019. L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6389–6398.

[14] Noe Samano, Mengjie Zhou, and Andrew Calway. 2020. You are here: Geolocation by embedding maps and images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*. Springer, 502–518.

[15] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. 2018. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 1–36.

[16] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 2019. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12716–12725.

[17] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4938–4947.

[18] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 2011. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*. IEEE, 667–674.

[19] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. 2018. Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 536–551.

[20] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. 2020. Optimal feature transport for cross-view image geo-localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11990–11997.

[21] Tim Y Tang, Daniele De Martini, Shangzhe Wu, and Paul Newman. 2021. Self-supervised learning for using overhead imagery as maps in outdoor range sensor localization. *The International Journal of Robotics Research* 40, 12-14 (2021), 1488–1509.

[22] Ultralytics. 2023. Ultralytics GitHub Repository. https://github.com/ultralytics/ultralytics. Accessed: 2024.

[23] Dan Wang, Xinrui Cui, Xun Chen, Zhengxia Zou, Tianyang Shi, Septimiu Salcudean, Z Jane Wang, and Rabab Ward. 2021. Multi-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5722–5731.

[24] Yanhong Wang, Ruizhen Zhao, Liequan Liang, Xinwei Zheng, Yigang Cen, and Shichao Kan. 2021. Block-based image matching for image retrieval. *Journal of Visual Communication and Image Representation* 74 (2021), 102998.

[25] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian FP Kooij. 2022. Visual cross-view metric localization with dense uncertainty estimates. In *European Conference on Computer Vision*. Springer, 90–106.

[26] Sijie Zhu, Taojiannan Yang, and Chen Chen. 2021. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3640–3649.

Automating Bibliometric Analysis with Sentence Transformers and Retrieval-Augmented Generation (RAG): A Pilot Study in Semantic and Contextual Search for Customized Literature Characterization for High-Impact Urban Research

Haowen Xu
xuh4@ornl.gov
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Xueping Li
Xueping.Li@utk.edu
University of Tennessee, Knoxville
Knoxville, Tennessee, USA

Jose Tupayachi
jtupayac@vols.utk.edu
University of Tennessee, Knoxville
Knoxville, Tennessee, USA

Jianming (Jamie) Lian
lianj@ornl.gov
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Olufemi A Omitaomu
omitaomuoa@ornl.gov
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

ABSTRACT

Bibliometric analysis is essential for understanding research trends, scope, and impact in urban science, especially in high-impact journals, such Nature Portfolios. However, traditional methods, relying on keyword searches and basic NLP techniques, often fail to uncover valuable insights not explicitly stated in article titles or keywords. These approaches are unable to perform semantic searches and contextual understanding, limiting their effectiveness in classifying topics and characterizing studies. In this paper, we address these limitations by leveraging Generative AI models, specifically transformers and Retrieval-Augmented Generation (RAG), to automate and enhance bibliometric analysis. We developed a technical workflow that integrates a vector database, Sentence Transformers, a Gaussian Mixture Model (GMM), Retrieval Agent, and Large Language Models (LLMs) to enable contextual search, topic ranking, and characterization of research using customized prompt templates. A pilot study analyzing 223 urban science-related articles published in Nature Communications over the past decade highlights the effectiveness of our approach in generating insightful summary statistics on the quality, scope, and characteristics of papers in high-impact journals. This study introduces a new paradigm for enhancing bibliometric analysis and knowledge retrieval in urban research, positioning an AI agent as a powerful tool for advancing research evaluation and understanding.

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

UrbanAI'24, October 29-November 1 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1156-5/24/10…$15.00
https://doi.org/10.1145/3681780.3697252

KEYWORDS

Bibliometrics Analysis, Large Language Models, Retrieval-Augmented Generation, Transformers

ACM Reference Format:
Haowen Xu, Xueping Li, Jose Tupayachi, Jianming (Jamie) Lian, and Olufemi A Omitaomu. 2024. Automating Bibliometric Analysis with Sentence Transformers and Retrieval-Augmented Generation (RAG): A Pilot Study in Semantic and Contextual Search for Customized Literature Characterization for High-Impact Urban Research . In *2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI (UrbanAI'24), October 29-November 1 2024, Atlanta, GA, USA.* ACM, Seattle, WA, USA, 7 pages. https://doi.org/10.1145/3681780.3697252

1 INTRODUCTION

Bibliometric analysis is a widely used method for evaluating and mapping research trends, impact, and scope across various scientific domains [2]. It provides quantitative insights by analyzing publication records, citations, and other scholarly outputs, helping researchers and policymakers understand the evolution of specific fields [4]. Over the past few decades, bibliometric analysis has evolved from basic citation counts and keyword frequency metrics to more sophisticated approaches, incorporating co-authorship networks, citation flows, and research topic clusters [7]. These methods are particularly important in fields like urban science, where emerging topics such as smart cities require continuous monitoring to shape the direction of future research and innovation [5]. Bibliometric analysis plays a key role in identifying influential works, emerging themes, and research gaps, thus guiding strategic decision-making in urban science and smart city development [15].

However, traditional bibliometric methods face several limitations. Most rely heavily on keyword searches and basic text mining techniques, which depend on exact matches of terminologies and predefined keywords. These techniques often miss critical insights that are not explicitly captured in the titles or abstracts of research articles, thereby limiting the ability to fully understand and classify research topics [8]. Furthermore, traditional natural language processing (NLP) approaches, such as term frequency-inverse document frequency (TF-IDF) or simple word co-occurrence metrics, fail to capture the semantic meaning and contextual relationships between concepts[9]. Although topic modeling methods like Latent Dirichlet Allocation (LDA) can offer significant benefits for bibliometric analysis by providing deeper insights into the relationships and structures within research literature [1], they are primarily used for uncovering thematic structures and classifying article topics and are not designed for enabling semantic search or providing a contextual understanding of an article that involves deeper reasoning and interpretation. As a result, traditional bibliometric analysis often falls short in generating deeper insights that require a thorough review and interpretation of the article's full content. Relying primarily on the analysis of titles, keywords, and standard metadata, limits the ability to provide a more customized and nuanced characterization of research based on the full textual data.

Recent advancements in generative AI models, such as large language models (LLMs), have opened new opportunities for enhancing research [11]. These models, including transformers and Retrieval-Augmented Generation (RAG) systems, excel at semantic understanding and contextual interpretation of complex texts, making them highly suitable for extracting valuable insights from research articles and technical manuals [10, 14]. In the field of urban informatics, LLMs have been increasingly applied to analyze large volumes of text, uncovering patterns and trends that traditional methods would overlook [6, 13]. In this paper, we propose a novel technical workflow for automating and enhancing bibliometric analysis by integrating Vector Databases, Sentence Transformers, Gaussian Mixture Models (GMM), Retrieval Agents, and an LLM. Our approach enables contextual search, topic ranking, and customized characterization of research articles, which we demonstrate through a pilot study analyzing 201 urban science-related articles published in Nature Communications over the past decade. This work addresses the limitations of traditional bibliometric methods, introducing a new paradigm for urban research analysis and knowledge retrieval through the development of AI agents with contextual understanding and reasoning capabilities.

2 LITERATURE REVIEW

To overcome the limitations and knowledge gaps in traditional bibliometric analysis, recent studies have employed generative AI models, particularly transformer-based language models, to automate and enhance bibliometric methodologies.

Fijačko et al. [3] explores the application of generative AI in bibliometric analysis, focusing on 10 years of research abstracts from the European Resuscitation Congresses (ERC). Using ChatGPT-4, the study classified 2,491 abstracts into ERC guideline topics, with Basic Life Support and Adult Advanced Life Support being the most frequent. The research highlights the potential of large language models like ChatGPT-4 in categorizing and analyzing scientific literature and identifying trends. However, challenges included potential misclassification, the limited use of abstract titles rather than full-text, and heavy reliance on the model's capabilities. These constraints highlight the challenges of automating bibliometric analysis in the absence of comprehensive datasets. However, the study effectively showcases the potential of AI to significantly improve bibliometric methodologies despite these limitations.

Weng et al. [12] introduces a methodology for detecting and visualizing key research topics using GPT-3 embeddings and the HDBSCAN clustering algorithm on 593 abstracts related to urban studies and machine learning. By clustering abstracts based on semantic similarity and extracting keywords using the Maximal Marginal Relevance (MMR) algorithm, the study provides an interactive tool for exploring abstract clusters and their associated topics. Challenges included optimizing clustering parameters and relying solely on abstracts, which may not fully represent the research. Some clusters contained outliers or minimal data, affecting accuracy. Despite these limitations, the study demonstrates the potential of transformer-based models in facilitating unsupervised bibliometric analysis, though refinement is needed.

Both articles emphasize the benefits of transformer-based and large language models for bibliometric analysis, while also addressing critical limitations such as data quality, optimization challenges, and input constraints when working with abstract-based datasets. To overcome these challenges, there is a need to harness recent advancements in sentence transformer models and RAG technologies. These innovations can enable the development of an AI agent capable of advanced contextual understanding of research articles, facilitating semantic search and providing tailored insights based on user-specific queries. This, in turn, can generate new bibliometric metrics, offering deeper and more comprehensive analysis.

3 METHODOLOGY

This section starts by outlining the design requirements for our proposed methods, then presents the conceptual workflow and its implementation, which combines Generative AI techniques with statistical models.

3.1 Design Requirements

Overall, we aim to develop an AI-agent styled tool that can interact with users, who are primarily researchers and college students, through human nature conversations, to get their inquire on the current-state of cutting edge research in a specific domain, such as smart city and urban science. Based on the inquiry, our workflow will automate a sequence of procedures that leverage the unique capabilities of sentence transformers and RAG techniques on a batch of selected literature filtered and downloaded from academics databases, such as Scopous, IEEE Xplore, and Web of Science. Aiming to shed lights on more advanced, intelligent, and automonous biblimetric analysis, our workflow aims to enable the following features:

Conversational Interaction: A chatbot-style interface will be implemented, allowing users to ask questions through natural human conversations, without the need for pre-defined keywords or technical jargon. This feature will enable users to define search and filter criteria for subsets of bibliographic data (e.g., research articles, conference proceedings, technical reports, and manuals) that have been pre-selected and downloaded from popular academic databases. The search process will be guided by broad categories, such as domains, disciplines, and journals, to streamline access to relevant literature.

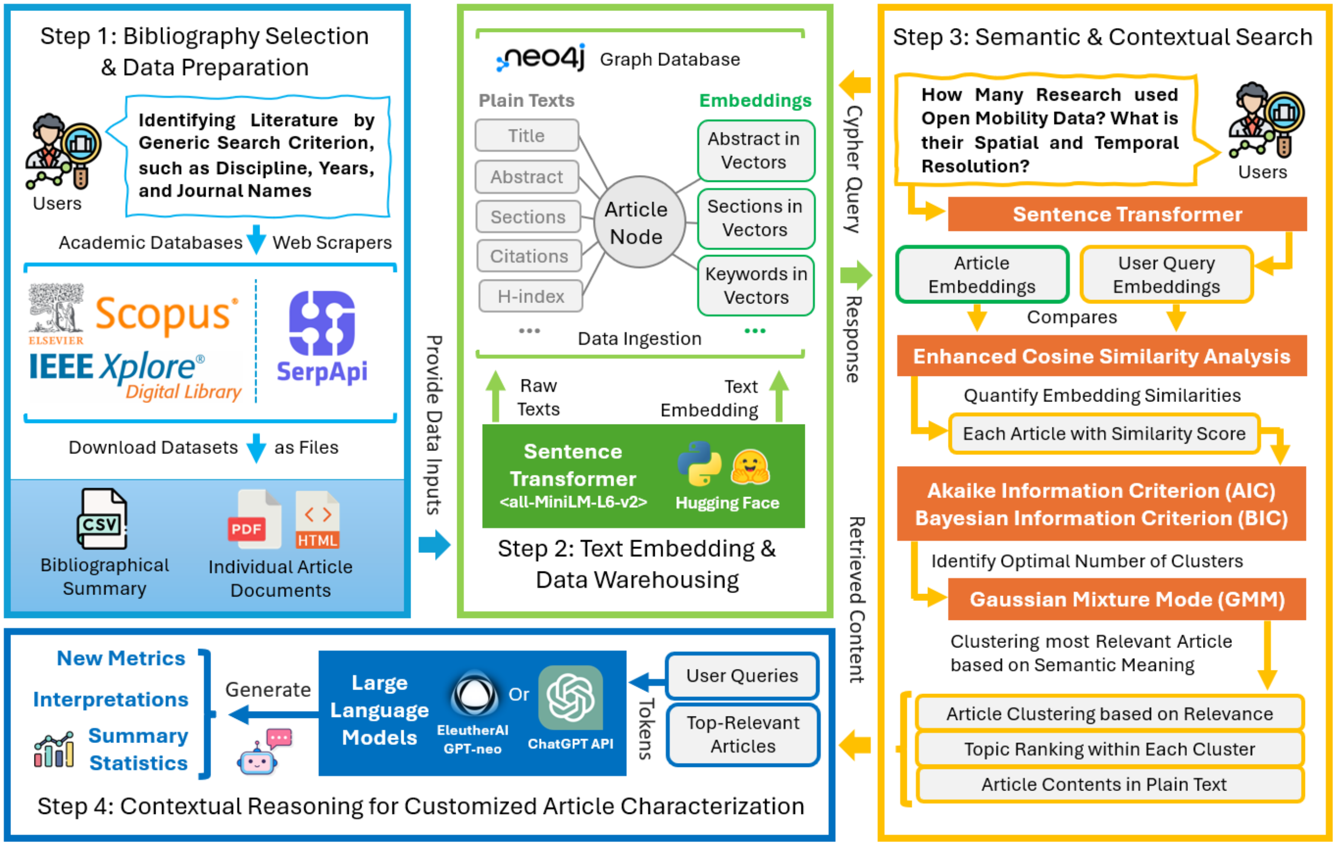


Figure 1: The overall design of the transformer and RAG-powered workflow.

Semantic and Contextual Search: Based on the user-defined inquiry, this process matches and retrieves relevant research documents or specific sections by analyzing the underlying meaning and contextual relationships between words, rather than relying solely on keyword matching. The use of sentence transformers and text embeddings, enables users to access information and knowledge based on conceptual relevance, rather than simple term frequency. This enhances the precision of literature filtering and facilitates deeper, more insightful knowledge discovery, which will plays important role as the retrieval agent within the RAG paradigm to benefit further analytics using Generative AI models.

Customized Literature Characterization: Using the output literature from the semantic and contextual search as input, Generative Pre-trained Transformer (GPT) models will be employed for contextual understanding, reasoning, and interpretation. These GPT models will process user inquiries to generate customized characterizations and interpretations of the selected literature, providing deeper insights and creating more sophisticated metrics for advanced bibliometric analyses. This approach aims to enhance the overall understanding of research trends and offer tailored, in-depth evaluations of the literature.

As an example of our end-user capability, a user could ask the chatbot, powered by our method, a question like, "What percentage of research published in Computers, Environment and Urban Systems over the past 5 years in the urban mobility sector uses traffic simulation-based methods versus crowd-sourced data-driven methods, and what are their spatial scales?" The semantic and contextual search then filters and retrieves relevant articles based on the query, ranks them by relevance, and feeds them to the generative AI model. This enables advanced contextual understanding and reasoning to provide customized characterizations on individual research's simulation types and spatial scales, which involve information often not found in keywords or titles. These characterizations can be later used to generate summary statics and insights to facilitate more detailed trend analysis and thematic mapping.

3.2 Workflow Design

Our workflow consists of four key procedures, as depicted in Figure 1. The workflow is later implemented in a Jupyter lab environment using Python-based libraries. Each procedure is detailed through the following following list.

3.2.1 Bibliography Selection and Data Preparation. In the first step, users select literature based on generic search criteria such as discipline, publication year, and journal name. Data is extracted from academic databases like Scopus, IEEE Xplore, and SerpApi using their respective web services and platforms, or through custom-built web scrapers, such as those powered by SerpAPI. The retrieved data includes bibliographic summaries in CSV format and individual articles in formats like PDF and HTML, which are then stored in a file-based system for further processing.

3.2.2 Text Embedding and Data Warehousing. After retrieving the essential documents, a Python script powered by PyPDF2 is used to parse the bibliographic summaries, which include the list of downloaded articles along with supportive metadata (e.g., authors, year, source, citations, and h-index), as well as the PDF and HTML versions of the individual articles. This parsing process is designed to upload key textual information into a datastore, building the knowledge base for the proposed AI agent. Unlike traditional information and content management systems, our workflow utilizes a sentence transformer, specifically the all-MiniLM-L6-v2 model from Hugging Face, to generate text embeddings—vector representations that encode the semantic and contextual meaning of the text. Compared with traditional NLP methods, sentence transformers, with its unique self-attention mechanism, have superior advantages in capturing semantic meaning, enabling contextual understanding, handling synonyms, and long-range dependencies between words in a sentence. These embeddings facilitate more efficient semantic and contextual searches in later stages of the workflow. The text embeddings, along with essential metadata and article content, are uploaded into the datastore. We selected Neo4j, a graph database, as the datastore for this workflow due to its graph data model, which better represents the relationships between data entities stored as nodes in the database. In our project, individual articles are represented as nodes within Neo4j, with associated metadata, content, and text embeddings stored as properties of each node.

3.2.3 Semantic and Contextual Search. In the third step, the workflow enables semantic and contextual searches within the literature stored in the knowledge base, leveraging the Neo4j database and sentence transformers. User queries, collected through a chatbot interface, serve as input for this advanced search. The core functionality compares the text embeddings of the user queries with those of the article contents. We employ an enhanced cosine similarity analysis, as described in Eq. 1, to calculate a similarity score ranging from 0 to 1, where 0 represents complete irrelevance and 1 represents high relevance. Our implementation extends the standard cosine similarity formula by using Python to chunk the original article content into sections and paragraphs, enabling more granular comparisons between the query and specific parts of the article. This process is applied to each article in the database, generating a similarity score based on semantic similarity with the user's query. At the contextual level, the framework evaluates the query's context and intelligently selects embeddings from different sections of the articles to perform a targeted and accurate search.

$$\text{Similarity Score} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \tag{1}$$

To draw the decision boundary based on a list of individual article's similarity score, we employed GMM to rank and cluster articles by their similarity score, which reflects their relevance. A GMM is a probabilistic model that represents a distribution of data as a mixture of multiple Gaussian (normal) distributions, each characterized by its own mean and variance, making it effective for modeling complex, multimodal datasets. We employed the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), alongside the elbow method, to determine the optimal number of clusters for the Gaussian Mixture Model (GMM) analysis. After the clustering analysis, the cluster with the highest average similarity scores implies it contains the most relevant articles, which are also further ranked based on its similarity score.

Articles in the top-ranked clusters are subsequently fed into generative AI models, specifically GPT, to enable more in-depth analysis and interpretation. The semantic and contextual search within this workflow is a critical component of the RAG paradigm, allowing for further subsetting and refining of input information to ensure more accurate and relevant results. This process also helps prevent exceeding the token limits of the GPT model context window by optimizing the selection of input texts.

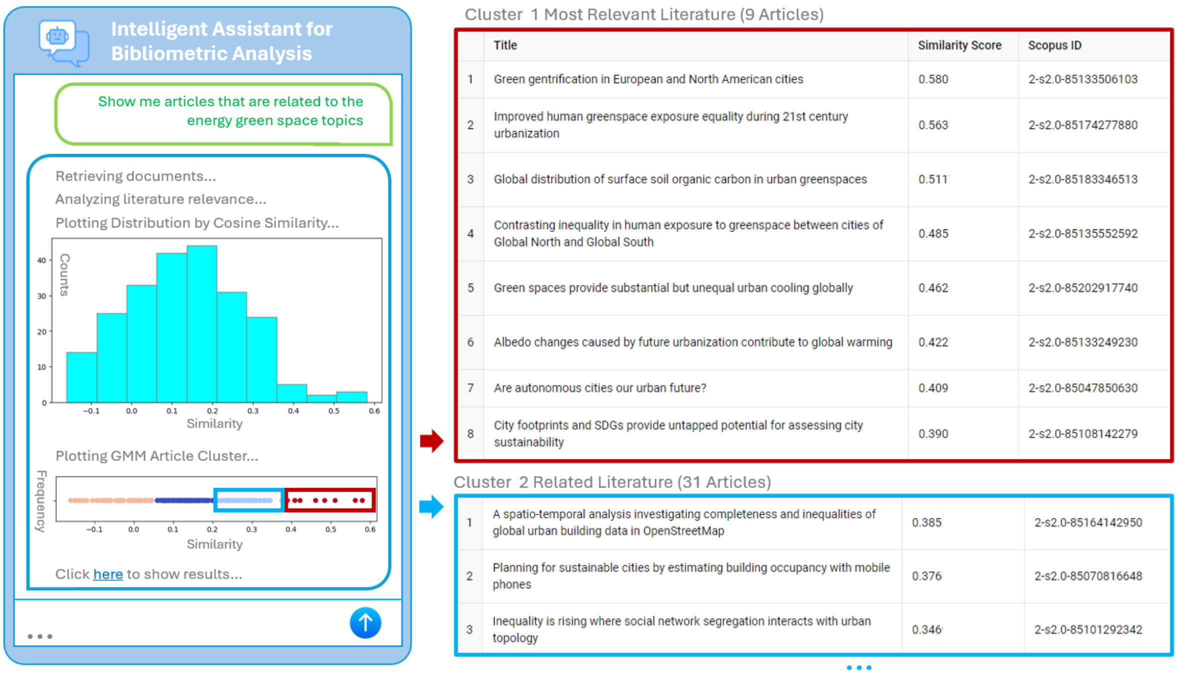
3.2.4 Customized Article Characterization. The top-ranked clusters, containing the most relevant articles, are then imported into a GPT model as an external knowledge source to generate customized characterizations for each article. These tailored bibliographic characteristics serve as metrics, providing more detailed descriptions and classifications of the articles. This approach uncovers valuable insights into research trends, focusing on individual articles' topics, technologies, methods, and contributions.

We leverage the contextual reasoning capabilities of large language models to classify and justify findings based on the semantic meaning of sections and paragraphs within the articles, extracting useful information without relying on precise names or keywords. This process is guided by instructional prompting strategies, where we design engineered prompt templates and feed them into the GPT model along with the relevant article text segments (specific sections). These segments are further refined and filtered based on their content relevance to ensure accurate classification and extraction.

At the technical level, we explored and tested the capabilities of two GPT models, including a local instance of EleutherAI/gpt-neo-1.3B models and the ChatGPT-3.5 Turbo API. Our experiments reveals that small models with on 1.3B parameters suffer from severe hallucination, and are unable to analyze large size of tokens. The ChatGPT-3.5 API demonstrates stable performance, particularly in its ability to process large text segments efficiently and produce reasoned characterizations.

4 PILOT STUDY

This pilot study aims to demonstrate the feasibility and performance of our proposed methods. For this study, we compiled a dataset of 223 high-impact urban research articles published in Nature Communications, obtained through the following Scopus query: TITLE-ABS-KEY ("smart city" OR "urban" OR "urban management" OR "urban planning") AND SRCTITLE ("Nature Communications") AND PUBYEAR > 2013. We preprocessed the dataset by removing all intermediate versions labeled as "Author Correction" or "Publisher Correction." The final dataset consists of a CSV file containing bibliometric summaries with all Scopus fields selected, along with 223 individual PDF documents of the actual articles.



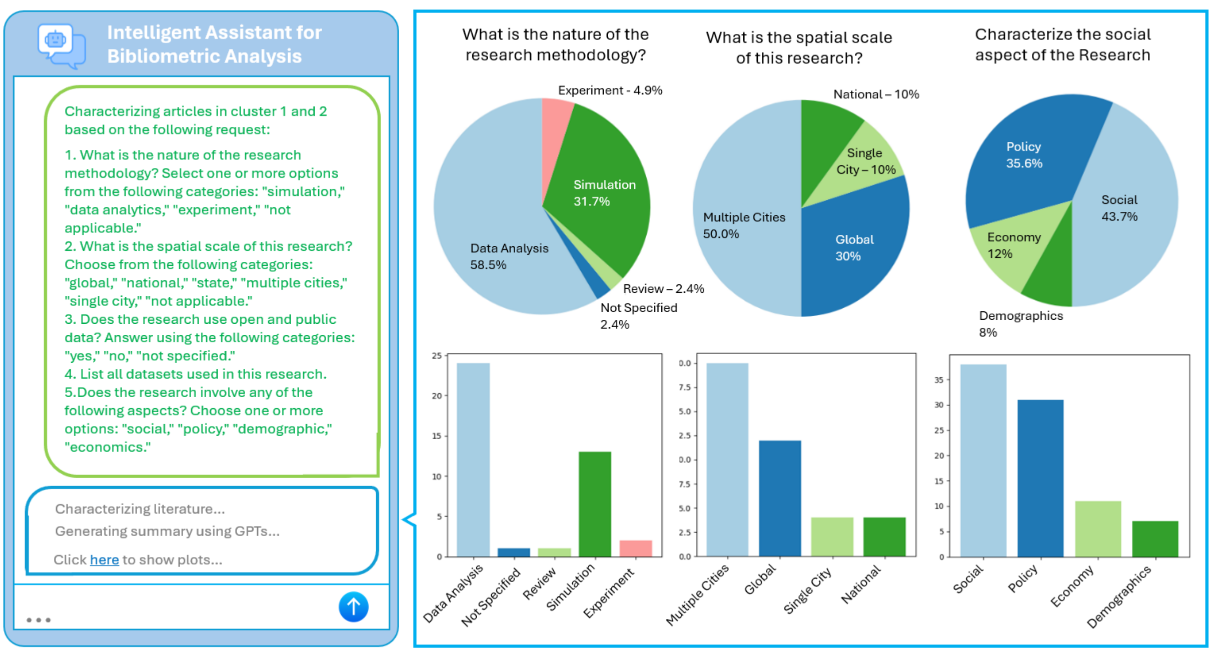
Cluster 1 Most Relevant Literature (9 Articles)

	Title	Similarity Score	Scopus ID
1	Green gentrification in European and North American cities	0.580	2-s2.0-85133506103
2	Improved human greenspace exposure equality during 21st century urbanization	0.563	2-s2.0-85174277880
3	Global distribution of surface soil organic carbon in urban greenspaces	0.511	2-s2.0-85183346513
4	Contrasting inequality in human exposure to greenspace between cities of Global North and Global South	0.485	2-s2.0-85135552592
5	Green spaces provide substantial but unequal urban cooling globally	0.462	2-s2.0-85202917740
6	Albedo changes caused by future urbanization contribute to global warming	0.422	2-s2.0-85133249230
7	Are autonomous cities our urban future?	0.409	2-s2.0-85047850630
8	City footprints and SDGs provide untapped potential for assessing city sustainability	0.390	2-s2.0-85108142279

Cluster 2 Related Literature (31 Articles)

1	A spatio-temporal analysis investigating completeness and inequalities of global urban building data in OpenStreetMap	0.385	2-s2.0-85164142950
2	Planning for sustainable cities by estimating building occupancy with mobile phones	0.376	2-s2.0-85070816648
3	Inequality is rising where social network segregation interacts with urban topology	0.346	2-s2.0-85101292342

(a) Semantic and contextual search based on user's query.



(b) Customized Literature Characterization using GPT's reasoning capability.

Figure 2: Demonstration of the workflow through two use case.

4.1 Use Case Demonstration

Our first use case on semantic and contextual search is demonstrated in Figure 2a, where the user submits an inquiry to identify articles related to urban green space. The chatbot responds by visualizing a histogram of similarity scores for all articles and displaying the GMM clusters of the articles. Additionally, a link is provided to download a CSV file that ranks and clusters the articles based on their relevance to the user's query.

Our second use case builds on the output articles from the first use case and demonstrates the capability to generate customized literature characterizations, creating new metrics for bibliometric analysis. Through the chatbot interface, users specify requests and instructions via prompts to guide the GPT model in generating tailored metrics. Examples of these prompts are illustrated in Figure 2b. Based on the prompts, the GPT processes the 67 retrieved articles and their critical content, leveraging its contextual reasoning ability to derive new literature characteristics, which can then be developed into metrics and summary statistics. Figure 2b also visualizes the responses to the user's queries using pie charts and box plots. Users can submit additional questions and custom requests through the chatbot to extract information and develop unique metrics tailored to the needs of bibliometric analysis.

Our major contribution lies in the development of an autonomous AI agent designed to assist researchers in automating the characterization and information extraction of large volumes of literature, including datasets exceeding 1,000 articles. This system enables the generation of in-depth insights for bibliometric analysis, significantly enhancing the scalability and depth of literature review and research trend identification processes. By automating these tasks, the AI agent offers a powerful tool for efficiently managing and analyzing extensive collections of scholarly articles, ultimately facilitating more comprehensive and insightful bibliometric analyses.

4.2 Limitation and Future Work

Developed as a prototype for a more advanced knowledge base and management system, our workflow still faces a few limitations, as the following:

- **Token Size Limitation:** The current implementation using the ChatGPT API has a maximum token size limitation and incurs service fees based on the number of tokens processed. This makes it less suitable for analyzing large volumes of literature.
- **Database Query Performance:** The current datastore implementation using the Neo4j database may encounter challenges in querying and managing large volumes of embedding data, as Neo4j is not optimized as a dedicated vector database.
- **Lack of Evaluation and Validation:** The GPT-generated literature characteristics are not currently evaluated by human experts, which introduces uncertainty regarding their accuracy and reliability.

As future work to address these limitations, we propose several experimental solutions. These include (a) deploying a local version of large language models, such as GPT-Neo, to minimize service fees for large-scale data analysis, (b) fine-tuning the GPT model to reduce unnecessary content and instructions sent to the model, thereby mitigating token size limitations, (c) transitioning our datastore implementation to dedicated vector databases, such as FAISS or Pinecone, to enhance latency and accuracy, and (d) developing a comprehensive strategy to evaluate the GPT's performance in analyzing and characterizing literature. Additionally, more advanced bibliometric analysis methods could be integrated into the current workflow to extend its analytical capabilities.

5 CONCLUSION

In this paper, we have presented a novel workflow that integrates generative AI models and advanced analytical techniques through the RAG paradigm to address the limitations of traditional bibliometric analysis methods. By leveraging the contextual reasoning capabilities of large language models and enhanced semantic search techniques, our system offers a more nuanced and insightful analysis of research literature. This approach, demonstrated through the analysis of urban science-related articles, enables customized characterizations and generates new metrics for bibliometric analysis, providing deeper insights into research trends, methodologies, and contributions.

Our pilot study demonstrates the feasibility of this workflow, showcasing its ability to facilitate advanced semantic and contextual searches, cluster relevant articles, and produce tailored bibliographic insights through generative AI. However, the current implementation faces challenges, including token size limitations, database query performance issues, and the lack of expert evaluation for the AI-generated results.

To address these limitations, future work will explore the deployment of local language models, fine-tuning of GPT models to optimize token usage, and transitioning to vector databases like FAISS or Pinecone to improve performance. Additionally, we aim to establish a comprehensive validation framework involving human experts to ensure the accuracy and reliability of the generated bibliometric insights. As advancements in AI and bibliometric methodologies continue, our workflow has the potential to serve as a powerful and autonomous tool for researchers and policymakers seeking to analyze and interpret vast bodies of scientific literature more effectively.

6 ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy (U.S DOE), Advanced Research Projects Agency–Energy (ARPA-E) under the project #DE-AR0001780. We thank our collaborators from the University of Tennessee Knoxville.

REFERENCES

[1] Xieling Chen and Haoran Xie. 2020. A structural topic modeling-based bibliometric study of sentiment analysis literature. *Cognitive Computation* 12 (2020), 1097–1129.

[2] Naveen Donthu, Satish Kumar, Debmalya Mukherjee, Nitesh Pandey, and Weng Marc Lim. 2021. How to conduct a bibliometric analysis: An overview and guidelines. *Journal of business research* 133 (2021), 285–296.

[3] Nino Fijačko, Ruth Masterson Creber, Benjamin S Abella, Primož Kocbek, Špela Metličar, Robert Greif, and Gregor Štiglic. 2024. Using generative artificial intelligence in bibliometric analysis: 10 years of research trends from the European Resuscitation Congresses. *Resuscitation Plus* 18 (2024), 100584.

[4] Ye-na Gan, Duo-duo Li, Nicola Robinson, and Jian-ping Liu. 2022. Practical guidance on bibliometric analysis and mapping knowledge domains methodology–A summary. *European Journal of Integrative Medicine* 56 (2022), 102203.

[5] Yi-Ming Guo, Zhen-Ling Huang, Ji Guo, Hua Li, Xing-Rong Guo, and Mpeoane Judith Nkeli. 2019. Bibliometric analysis on smart cities research. *Sustainability* 11, 13 (2019), 3606.

[6] Jianyuan Liang, Anqi Zhao, Shuyang Hou, Fengying Jin, and Huayi Wu. 2024. A GPT-enhanced framework on knowledge extraction and reuse for geographic analysis models in Google Earth Engine. *International Journal of Digital Earth* 17, 1 (2024), 2398063.

[7] Luis Javier Cabeza Ramírez, Sandra M Sánchez-Cañizares, and Fernando J Fuentes-García. 2019. Past themes and tracking research trends in entrepreneurship: A co-word, cites and usage count analysis. *Sustainability* 11, 11 (2019), 3121.

[8] Rodrigo Romero-Silva and Sander De Leeuw. 2021. Learning from the past to shape the future: A comprehensive text mining analysis of OR/MS reviews. *Omega* 100 (2021), 102388.

[9] Iqra Safder and Saeed-Ul Hassan. 2019. Bibliometric-enhanced information retrieval: a novel deep feature engineering approach for algorithm searching from full-text publications. *Scientometrics* 119 (2019), 257–277.

[10] Jose Tupayachi, Haowen Xu, Olufemi A Omitaomu, Mustafa Can Camur, Aliza Sharmin, and Xueping Li. 2024. Towards Next-Generation Urban Decision Support Systems through AI-Powered Construction of Scientific Ontology Using Large Language Models—A Case in Optimizing Intermodal Freight Transportation. *Smart Cities* 7, 5 (2024), 2392–2421.

[11] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science* 18, 6 (2024), 186345.

[12] Min-Hsien Weng, Shaoqun Wu, and Mark Dyer. 2022. Identification and visualization of key topics in scientific publications with transformer-based language models and document clustering methods. *Applied Sciences* 12, 21 (2022), 11220.

[13] Haowen Xu, Femi Omitaomu, Soheil Sabri, Sisi Zlatanova, Xiao Li, and Yongze Song. 2024. Leveraging Generative AI for Urban Digital Twins: A Scoping Review on the Autonomous Generation of Urban Data, Scenarios, Designs, and 3D City Models for Smart City Advancement. *arXiv preprint arXiv:2405.19464* (2024). https://doi.org/10.48550/arXiv.2405.19464 arXiv:2405.19464 [cs.AI] Computer Science > Artificial Intelligence.

[14] Haowen Xu, Jinghui Yuan, Anye Zhou, Guanhao Xu, Wan Li, Xinyue Ye, et al. 2024. GenAI-powered Multi-Agent Paradigm for Smart Urban Mobility: Opportunities and Challenges for Integrating Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) with Intelligent Transportation Systems. *arXiv preprint arXiv:2409.00494* (2024).

[15] Li Zhao, Zhi-ying Tang, and Xin Zou. 2019. Mapping the knowledge domain of smart-city research: A bibliometric and scientometric analysis. *Sustainability* 11, 23 (2019), 6648.

Encryption Techniques for Privacy-Preserving CNN Models: Performance and Practicality in Urban AI Applications

Sanaullah*
University of Applied Sciences and Arts
Bielefeld, Germany

Hasina Attaullah
University of Applied Sciences and Arts
Bielefeld, Germany

Thorsten Jungeblut
University of Applied Sciences and Arts
Bielefeld, Germany

Abstract

In recent years, as urban AI applications increasingly rely on sensitive data, ensuring the privacy and security of machine learning (ML) models has become essential. The proposed research study evaluates the performance and security trade-offs of seven encryption techniques applied to ML models used in urban AI settings. We compare encryption methods, including mixed homomorphic encryptions using Convolutional Neural Networks (CNNs) trained on the MNIST dataset, we analyze how these encryption methods affect model performance in terms of accuracy, error rate, and information leakage. The CNN models, after being trained with encrypted data, are deployed on embedded devices to evaluate real-time performance. We measure key metrics, including execution time, memory usage, and classification accuracy, to assess the feasibility of each encryption method in urban AI scenarios. Additionally, the impact of encryption on model interpretability and robustness is considered, particularly when used in urban applications like intelligent transportation systems, smart city sensors, and surveillance systems. By evaluating error rates, mutual information scores, and statistical properties such as mean and variance, this research aims to explore the practical trade-offs between security, privacy, and performance. Our findings highlight the importance of selecting appropriate encryption techniques for urban AI tasks to maintain both data privacy and model efficiency in real-world settings.

CCS Concepts

• **Security and privacy → Domain-specific security and privacy architectures**.

Keywords

Convolutional Neural Networks, Artificial Intelligence, Data Privacy, Encryption Techniques, Embedded Devices, Privacy-Preserving AI, Urban AI Applications

ACM Reference Format:
Sanaullah, Hasina Attaullah, and Thorsten Jungeblut. 2024. Encryption Techniques for Privacy-Preserving CNN Models: Performance and Practicality in Urban AI Applications. In *2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI (UrbanAI'24), October 29-November 1 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681780.3697244

UrbanAI'24, October 29-November 1 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1156-5/24/10
https://doi.org/10.1145/3681780.3697244

1 Introduction

In the age of data-driven decision-making advanced ML-based models are essential to urban AI applications, ranging from smart traffic management to public health monitoring. Ensuring the confidentiality and integrity of sensitive information in these contexts is critical, especially given the vast amounts of data generated by IoT devices and urban sensors [9, 19, 24]. However, this dependency raises significant concerns about data privacy and security [3, 14]. Increasing incidents of data breaches and unauthorized access, emphasize the urgent need to address these privacy concerns [16]. Thus, ML models in urban environments face two key challenges: protecting data confidentiality while maintaining performance. Traditional methods fall short, prompting interest in advanced encryption to secure data without harming ML functionality [12, 20, 23, 30]. For example, symmetric and asymmetric encryption can secure data in transit but often add computational overhead, which slows model performance during training and inference. [7, 28]. Additionally, data anonymization techniques are not sufficient against advanced attacks like model inversion, which can expose sensitive information and other advanced mathematical-based neural models [21, 25]. This gap between the demand for robust data privacy and the limitations of traditional techniques has driven the exploration of more sophisticated encryption methods.

Techniques like homomorphic encryption, differential privacy, and SMPC allow computations on encrypted data or add controlled noise, providing effective solutions. [2]. However, the effectiveness and efficiency of these methods in the context of modern ML applications, particularly in urban settings, remain largely underexplored [15]. Therefore, this study analyzes encryption methods for ML models to address these concerns, focusing on CNNs for image processing and classification. It evaluates how different techniques affect the performance, resource use, and practical usability of CNN training with a benchmark in image processing, such as the MNIST dataset [4]. This study not only evaluates encryption methods in standard approaches but also explores their performance on embedded devices such as the Jetson Nano, Jetson Orin Nano, Jetson Xavier NX, and Jetson Orin AGX [17, 26]. These devices are commonly used in edge computing and urban AI applications due to their compact size and specialized hardware for ML tasks [18, 27]. In addition to evaluating the encryption methods in a standard computational environment, this study also explores the performance of encrypted CNN models on various embedded devices, including the Jetson Nano, Jetson Orin Nano, Jetson Xavier NX, and Jetson Orin AGX [11, 17, 26]. These devices are commonly used in edge computing and urban AI applications due to their compact size and specialized hardware for ML tasks [18, 27]. This analysis evaluates encryption techniques on resource-constrained platforms to offer practical understanding for developers and researchers working on secure, privacy-preserving AI in urban applications.

2 Proposed Study

The proposed study addresses the challenge of balancing data protection with the practical needs of ML models, such as accuracy, efficiency, and usability in urban AI applications [5, 8, 29]. Therefore these approaches are structured into phases: dataset preparation, model training, encryption implementation, performance assessment, and result analysis. Focusing on urban AI and embedded devices, the study examines trade-offs between execution time, accuracy, and power consumption. Real-world case studies, like smart traffic management and public health surveillance, demonstrate how encryption can enhance data privacy while maintaining effectiveness in resource-constrained environments.

2.1 Encryption Methods

This study evaluates and compares the practical performance and security trade-offs of seven encryption techniques applied to CNN models in urban AI applications. Unlike previous research, which often remains theoretical, we implement XOR, differential privacy, secure multiparty computation (SMPC), attribute-based encryption (ABE), and various homomorphic encryption methods, including simple homomorphic encryption, fully homomorphic encryption (FHE), and Homomorphic Encryption for Arithmetic of Approximate Numbers (HEAAN) on real datasets. We assess their effects on model accuracy, error rate, and information leakage using CNNs trained on encrypted MNIST data. The primary objectives are to evaluate how these encryption methods influence the image processing and classification accuracy of CNN models trained on encrypted datasets compared to those trained on the original, unencrypted data [6, 10].

2.2 Embedded Devices

By deploying the encrypted models on embedded devices like the NVIDIA Jetson family (such as the Jetson Nano, Jetson Orin Nano, Jetson Xavier NX, and Jetson Orin AGX), we provide concrete benchmarks, measuring execution time, memory usage, and classification accuracy. Additionally, we analyze how these encryption methods impact model interpretability, robustness, and statistical properties like mutual information, mean, and variance, offering a detailed comparison of their practical use in urban AI tasks. This practical evaluation stands apart from prior studies by focusing not only on encryption efficacy but also on real-world deployment metrics, providing a deeper understanding of the balance between privacy, security, and model performance in sensitive urban AI applications.

2.3 Model Training and Evaluation

In this study, we implement a CNN model to classify the MNIST dataset of handwritten digits. Images are normalized to [0, 1] and reshaped to (28, 28, 1), with labels one-hot encoded. The CNN has two convolutional layers (32 and 64 filters), max-pooling layers, a flattening layer, and two dense layers, ending with a softmax output. The model uses the Adam optimizer and categorical cross-entropy for multi-class classification. Furthermore, the model is trained for 50 epochs with a batch size of 64, utilizing a validation set to monitor performance during training. Each encryption technique is then applied to both the training and testing datasets. Additionally, the proposed architecture evaluates the impact of different encryption techniques on the performance and error rates of the CNNs model trained on the MNIST dataset. Unlike traditional models that prioritize accuracy, this study examines how each encryption method affects learning from encrypted data. By measuring test accuracy, error rates, and statistical properties like mean and variance, it offers a detailed understanding of information leakage and model robustness in encrypted conditions—an area that has been underexplored in encrypted ML. This research bridges the gap between privacy-preserving techniques and high-performance models, particularly in urban AI applications dealing with public data [1, 13, 22]. Most state-of-the-art models focus on accuracy or differential privacy but fail to systematically explore multiple encryption schemes and their trade-offs within a unified CNN framework.

Table 1: Accuracy loss, and the error rate of the proposed models' encrypted dataset with different techniques.

Encryption Method	Accuracy	Loss	Error Rate
Unencrypted (Baseline)	0.9913	0.0279	0.0087
Attribute-Based Encryption (ABE)	0.9898	0.0331	0.0102
Homomorphic Encryption (HEAAN)	0.9875	0.0369	0.0125
Fully Homomorphic Encryption (FHE)	0.9900	0.0300	0.0100
Differential Privacy	0.7990	0.2989	0.0710

3 Experimental Results

We conducted experiments to evaluate the performance of CNN models trained on the MNIST dataset under different encryption techniques on different embedded devices to highlight the importance of these encryption techniques in the real-world implementation of urban AI applications. The results were compiled from multiple Jetson devices, focusing on model accuracy, error rates, and statistical properties (such as execution time) of the encrypted datasets using different encryption techniques.

3.1 Accuracy and Loss Comparison

The experimental results demonstrate a thorough evaluation of mixed encryption techniques and their impact on the performance of CNNs trained on the MNIST dataset. ABE and HEAAN emerged as the best-performing methods, achieving accuracies of 0.9898 and 0.9875, respectively. The effectiveness of these techniques can be attributed to their sophisticated algorithms that maintain data structure and relationships during encryption. ABE's ability to decrypt based on specific attributes ensures that relevant information remains intact for model training, while HEAAN's optimization for approximate arithmetic allows robust calculations on encrypted data, enhancing the learning process without significantly distorting the original data.

In contrast, Differential Privacy showed the lowest performance with an accuracy of 0.7990. This reduction can be attributed to the introduction of noise essential for enhancing privacy, which adversely affects the model's ability to learn from the original data. The added noise obscures meaningful patterns, resulting in less

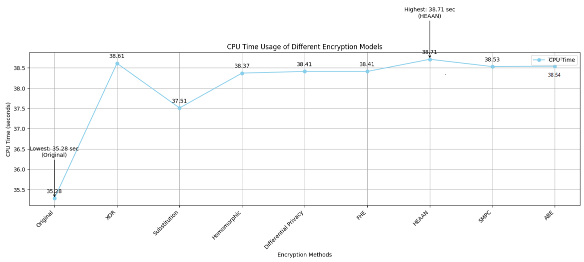


Figure 1: A graphical comparison of CPU time usage across the different encryption techniques, highlighting the resource demands associated with each method.

Table 2: Execution Time and Resource Utilization of Encrypted Models on Embedded Devices

Embedded Device	Execution Time (seconds)	System Utilization (%)	Power Consumption (mW)
Jetson Nano	28	78	5536
Jetson Xavier NX	20	63	5036
Jetson Orin Nano	18	57	23000
Jetson Orin AGX	8	36	9000

effective training and a diminished capacity to generalize from the dataset. On the other hand, FHE provided a balanced trade-off with an accuracy of approximately 0.9900. It allows for computations on encrypted data without exposing sensitive information, serving as a middle ground. However, it does not fully use the potential of more specialized methods like ABE or HEAAN, which are designed specifically for tasks requiring both data security and high model performance.

Furthermore, Table 1 presents an analysis of the accuracy and loss for CNN models trained on datasets encrypted with different techniques. This visual representation underscores the comparative effectiveness of each method. Additionally, Figure Table 1, 3rd column illustrates the error rates associated with each encryption technique, providing a comprehensive understanding of model reliability. Overall, the results indicate that while some encryption techniques maintain high accuracy and model efficacy, others, like Differential Privacy, introduce trade-offs that hinder performance. This analysis of each method's strengths and weaknesses emphasizes the importance of selecting appropriate encryption techniques based on specific urban AI application requirements, such as smart city monitoring, autonomous vehicle navigation, and public safety systems. In these contexts, ensuring data privacy while maintaining high model performance is crucial for fostering trust and reliability in AI-driven solutions.

3.2 Execution Time on Different Embedded Devices

The execution time for running CNN models with various encryption methods is crucial for assessing their feasibility in Urban AI applications, particularly regarding public data safety. The baseline model, operating without encryption, achieved an execution time of 35.28 seconds. In contrast, the XOR encryption method exhibited the least impact on execution time, requiring 38.61 seconds. This minor increase is due to the straightforward nature of XOR encryption, which incurs minimal computational overhead. More complex encryption techniques, such as FHE, Differential Privacy, and HEAAN, showed significant increases in execution times. FHE had an execution time of 38.41 seconds, while HEAAN took 38.71 seconds. These longer execution times can be attributed to the intricate operations involved in preserving data privacy while enabling computations on encrypted data. Figure 1 demonstrates a graphical comparison of CPU time usage across the different encryption techniques, highlighting the resource demands associated with each method. Additionally, the analysis of balanced FHE-encrypted models includes practical demonstrations across different embedded Jetson devices, demonstrating significant performance differences during real-time execution—critical for applications requiring immediate predictions. Key metrics such as execution time, overall system utilization, and power consumption were evaluated across the devices, as shown in Table 2.

On the Jetson Nano, the longer execution time of approximately 28 seconds, along with a system utilization of 78% and power consumption of 5536 mW, underscores the device's limitations in handling the computational demands of complex encrypted models. The Nano, while suitable for basic tasks, struggles with the intensive calculations required by FHE encryption, resulting in slower processing speeds and higher resource consumption. In contrast, the Jetson Xavier NX demonstrated improved performance with an execution time of 20 seconds, utilizing 63% of its system resources while consuming 5036 mW. This device's enhanced processing power allows it to perform more complex computations efficiently, but the higher power consumption reflects a trade-off between performance and energy efficiency. It showcases how advanced hardware can reduce execution time while still demanding significant power for high-performance tasks.

Similarly, the Jetson Orin Nano further optimized execution time to 18 seconds, despite consuming higher power at 23000 mW. This increased power consumption can be attributed to its advanced architecture, which is capable of efficiently managing more intricate computations. This allows the Orin Nano to perform better while requiring greater energy, indicating its design prioritizes performance for demanding applications. Finally, the Jetson Orin AGX excelled with an execution time of just 8 seconds, utilizing only 36% of its system resources while consuming 9000 mW. This exceptional performance highlights its superior processing capabilities, enabling it to efficiently handle encrypted models with minimal delay. Its ability to maintain lower system utilization while delivering high performance reflects an optimized balance between processing power and resource management. Therefore, the execution times of encrypted models on different embedded devices highlight the unstable computational demands associated with encryption techniques. While FHE and other encryption methods are vital for ensuring data security, their impact on execution efficiency is significant. Choosing the appropriate device is essential for optimizing performance and power consumption in Urban AI applications, where quick and secure processing of public data is critical.

4 Conclusion

In conclusion, this analysis thoroughly examines various encryption techniques applied to CNNs for image processing and classification tasks using the MNIST dataset. The findings demonstrate that the FHE technique and HEAAN offer the most effective balance between model accuracy and error rates, with FHE achieving an accuracy of 0.9874 and HEAAN slightly outperforming it at 0.9875. These encryption methods successfully maintain data privacy while ensuring robust model performance, making them particularly valuable in contexts where sensitive information is handled. Furthermore, the evaluation of CPU time usage demonstrates that the introduction of these encryption methods has a minimal impact on execution time, indicating their viability for practical applications. Assessments of power consumption and memory usage across various Jetson devices highlight that the Jetson Orin AGX stands out for its efficient performance in execution time and memory management, despite its higher power consumption. Therefore, these understandings are important for urban AI applications, where real-time processing of sensitive data is paramount. The ability to deploy encrypted datasets on embedded devices without compromising performance or privacy positions FHE and HEAAN as leading techniques in this field. Future research should aim to optimize these encryption methods further to enhance their computational efficiency and scalability. Additionally, exploring hybrid approaches that uses the strengths of multiple encryption techniques could offer a promising pathway to better balance privacy, performance, and resource consumption in urban AI systems. Therefore, by advancing these encryption strategies, we can provide new directions for safer and more efficient urban AI applications, ensuring that sensitive public data remains protected while harnessing the power of advanced ML algorithms.

5 Acknowledgement

This research was supported by the research training group "Dataninja" (Trustworthy AI for Seamless Problem Solving: Next Generation Intelligence Joins Robust Data Analysis) funded by the German federal state of North Rhine-Westphalia. This project is also supported by SAIL (grant no NW21-059B), a project funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia.

References

[1] Shahab S Band, Sina Ardabili, Mehdi Sookhak, Anthony Theodore Chronopoulos, Said Elnaffar, Massoud Moslehpour, Mako Csaba, Bernat Torok, Hao-Ting Pai, and Amir Mosavi. 2022. When smart cities get smarter via machine learning: An in-depth literature review. *IEEE Access* 10 (2022), 60985–61015.

[2] Amine Boulemtafes, Abdelouahid Derhab, and Yacine Challal. 2020. A review of privacy-preserving techniques for deep learning. *Neurocomputing* 384 (2020), 21–45.

[3] Dunren Che, Mejdl Safran, and Zhiyong Peng. 2013. From big data to big data mining: challenges, issues, and opportunities. In *Database Systems for Advanced Applications: 18th International Conference, DASFAA 2013, International Workshops: BDMA, SNSM, SeCoP, Wuhan, China, April 22-25, 2013. Proceedings 18*. Springer, 1–15.

[4] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2921–2926.

[5] Josep Domingo-Ferrer, Oriol Farras, Jordi Ribes-González, and David Sánchez. 2019. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Computer Communications* 140 (2019), 38–60.

[6] Ehsan, Hassan Takabi, and Mehdi Ghasemi. 2019. Deep neural networks classification over encrypted data. In *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*. 97–108.

[7] Adam J Elbirt. 2002. *Reconfigurable computing for symmetric-key algorithms*. Worcester Polytechnic Institute.

[8] Mohammed A Fadhel, Ali M Duhaim, Ahmed Saihood, Ahmed Sewify, Mokhaled NA Al-Hamadani, AS Albahri, Laith Alzubaidi, Ashish Gupta, Sayedali Mirjalili, and Yuantong Gu. 2024. Comprehensive systematic review of information fusion methods in smart cities and urban environments. *Information Fusion* (2024), 102317.

[9] Arash Heidari, Nima Jafari Navimipour, and Mehmet Unal. 2022. Applications of ML/DL in the management of smart cities and societies based on new trends in information technologies: A systematic literature review. *Sustainable Cities and Society* 85 (2022), 104089.

[10] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. 2017. Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189* (2017).

[11] Thorsten Jungeblut. [n. d.]. A Spike Vision Approach for Multi-object Detection and Generating Dataset Using Multi-core Architecture on Edge Device. In *Engineering Applications of Neural Networks: 25th International Conference, EANN 2024, Corfu, Greece, June 27–30, 2024, Proceedings*. Springer Nature, 317.

[12] Shamini Koravuna, Ulrich Rückert, Thorsten Jungeblut, et al. 2023. Evaluation of Spiking Neural Nets-Based Image Classification Using the Runtime Simulator RAVSim. *International Journal of Neural Systems* (2023), 2350044–2350044.

[13] Agostino Marengo. 2023. The Future of AI in IoT: Emerging Trends in Intelligent Data Analysis and Privacy Protection. (2023).

[14] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. 2016. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing* 2016 (2016), 1–16.

[15] Md Abdur Rahman, M Shamim Hossain, Ahmad J Showail, Nabil A Alrajeh, and Mohammed F Alhamid. 2021. A secure, private, and explainable IoHT framework to support sustainable health monitoring in a smart city. *Sustainable Cities and Society* 72 (2021), 103083.

[16] Alan Charles Raul. 2021. *The privacy, data protection and cybersecurity law review*. Law Business Research Limited.

[17] Kaushik Roy, Ulrich Ruckert, Thorsten Jungeblut, et al. 2024. Poster: Selection of Optimal Neural Model using Spiking Neural Network for Edge Computing. In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1452–1453.

[18] Milad Salem, Shayan Taheri, and Jiann-Shiun Yuan. 2018. Utilizing transfer learning and homomorphic encryption in a privacy preserving and secure biometric recognition system. *Computers* 8, 1 (2018), 3.

[19] Sanaullah, Hasan Baig, Jan Madsen, and Jeong-A Lee. 2020. A parallel approach to perform threshold value and propagation delay analyses of genetic logic circuit models. *ACS Synthetic Biology* 9, 12 (2020), 3422–3428.

[20] Sanaullah, Shamini Koravuna, Ulrich Rückert, and Thorsten Jungeblut. 2022. SNNs model analyzing and visualizing experimentation using RAVSim. In *International conference on engineering applications of neural networks*. Springer, 40–51.

[21] Sanaullah, Shamini Koravuna, Ulrich Rückert, and Thorsten Jungeblut. 2023. Exploring spiking neural networks: a comprehensive analysis of mathematical models and applications. *Frontiers in Computational Neuroscience* 17 (2023), 1215824.

[22] Sanaullah Sanaullah. 2024. A Hybrid Spiking-Convolutional Neural Network Approach for Advancing Machine Learning Models. In *Northern Lights Deep Learning Conference*. PMLR, 220–227.

[23] Sanaullah Sanaullah, Hasina Attaullah, and Thorsten Jungeblut. 2024. The Next-Gen Interactive Runtime Simulator for Neural Network Programming. In *Companion Proceedings of the 8th International Conference on the Art, Science, and Engineering of Programming*. 8–10.

[24] Sanaullah Sanaullah, Hasan Baig, and Jeong A Lee. 2019. Accelerating the Threshold and Timing Analysis of Genetic Logic Circuit Models. (2019).

[25] SK Sanaullah, U Rückert, and T Jungeblut. 2023. Design-space exploration of SNN models using application-specific multi-core architectures. In *The Neuro-Inspired Computing Elements Conference (NICE)*.

[26] Prashanthi SK, Sai Anuroop Kesanapalli, and Yogesh Simmhan. 2022. Characterizing the performance of accelerated jetson edge devices for training deep learning models. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 3 (2022), 1–26.

[27] Sana Ullah and Thorsten Jungeblut. 2023. Analysis of MR Images for Early and Accurate Detection of Brain Tumor using Resource Efficient Simulator Brain Analysis. In *19th International Conference on Machine Learning and Data Mining MLDM*.

[28] Yong Wang, Garhan Attebury, and Byrav Ramamurthy. 2006. A survey of security issues in wireless sensor networks. (2006).

[29] Yang Yang, Xindi Huang, Ximeng Liu, Hongju Cheng, Jian Weng, Xiangyang Luo, and Victor Chang. 2019. A comprehensive survey on secure outsourced computation and its applications. *IEEE Access* 7 (2019), 159426–159465.

[30] Umar Zaman, Imran, Faisal Mehmood, Naeem Iqbal, Jungsuk Kim, and Muhammad Ibrahim. 2022. Towards secure and intelligent internet of health things: A survey of enabling technologies and applications. *Electronics* 11, 12 (2022), 1893.

SurfaceAI: Automated creation of cohesive road surface quality datasets based on open street-level imagery

Alexandra Kapp, Edith Hoffmann, Esther Weigmann, Helena Mihaljević
Hochschule für Technik und Wirtschaft Berlin (HTW Berlin)
Berlin, Germany

Abstract

This paper introduces SurfaceAI, a pipeline designed to generate comprehensive georeferenced datasets on road surface type and quality from openly available street-level imagery. The motivation stems from the significant impact of road unevenness on the safety and comfort of traffic participants, especially vulnerable road users, emphasizing the need for detailed road surface data in infrastructure modeling and analysis. SurfaceAI addresses this gap by leveraging crowdsourced Mapillary data to train models that predict the type and quality of road surfaces visible in street-level images, which are then aggregated to provide cohesive information on entire road segment conditions.

CCS Concepts

• **Applied computing → Cartography**; • **Computing methodologies → Computer vision**.

Keywords

Road Surface Classification, Road Quality Classification, Deep Learning, Open Data, Street-Level Imagery, Mapillary, OpenStreetMap

ACM Reference Format:
Alexandra Kapp, Edith Hoffmann, Esther Weigmann, Helena Mihaljević. 2024. SurfaceAI: Automated creation of cohesive road surface quality datasets based on open street-level imagery. In *2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI (UrbanAI'24), October 29-November 1, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681780.3697277

1 Introduction

Road damages can have a significant impact on the comfort and safety of traffic participants, especially on vulnerable road users such as cyclists [9] or wheelchair users [20], and have been identified as a relevant cause for traffic accidents [17]. Thus, comprehensive information on road surface types (e.g., paving stones) and their quality is crucial for the analysis and advancement of road infrastructure [25], or for routing applications [12]. However, the required data sources are commonly lacking. While OpenStreetMap (OSM) offers tags for surface type and quality, large gaps within the database exist as data availability depends on contributions from volunteers. For example, as of August 2024, only 8,6% of road segments in Germany are tagged with quality information.

There is a large body of research on road surface state assessment [6, 16, 23], focusing primarily on detecting damages such as cracks and potholes [10, 27, 28]. Yet, road damage alone may not reflect the full range of factors that influence a traffic participant's experience. For example, the smoothness of sett (regular-shaped cobblestone) is also influenced by the flatness of the stones. Thus, a combination of road surface type and quality information is necessary to provide a comprehensive picture of the driving experience. Satellite images have been proposed for large-scale surface type identification [29], but limited to distinguishing between paved and unpaved roads. Karakaya et al. [15] used crowdsourced accelerometer data to derive surface quality information for bicycle paths and demonstrated its applicability in routing. While this approach can predict comfort, it can struggle with road type differentiation (e.g., roadway or sidewalk) in a crowdsourced setting, and relies on sensor data which is not widely available yet.

Rateke et al. [22] developed models predicting the surface type and quality from street-level imagery. The solution presented in this work incorporates image classification models of similar architecture. However, we extend their approach by utilizing a finer-grained category scheme, a more diverse dataset, and optimizations in the model architecture. Moreover, to the best of our knowledge, no existing works have presented a pipeline enabling the prediction of road surface type and quality on a large scale for comprehensive road networks across arbitrary municipalities using available data. We aim to fill this gap by presenting *SurfaceAI*, a process pipeline and a software that uses openly available street-level imagery to generate georeferenced datasets pertaining to surface type and quality across arbitrary areas.

2 Proposed approach

Street-level images are collected based on a user-defined geographic boundary of interest. These images are then classified by road type, surface type, and surface quality, and aggregated at the road segment level to provide estimates for the entire road network within the defined boundary (see Figure 1). The individual steps of the pipeline are elaborated below.

We implement the pipeline as open-source code in Python, using PyTorch [19] for neural network models and PostGIS [7] for efficient spatial operations[1].

2.1 Street-level imagery

Our approach leverages *Mapillary* [3], a crowdsourcing platform launched in 2013 that provides openly available street-level images. As of January 2024, the dataset contains about 170 million images in Germany, with over 50% captured within the past three years.

[1]Github Repository: https://github.com/SurfaceAI/road_network_classification

UrbanAI'24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1156-5/24/10
https://doi.org/10.1145/3681780.3697277

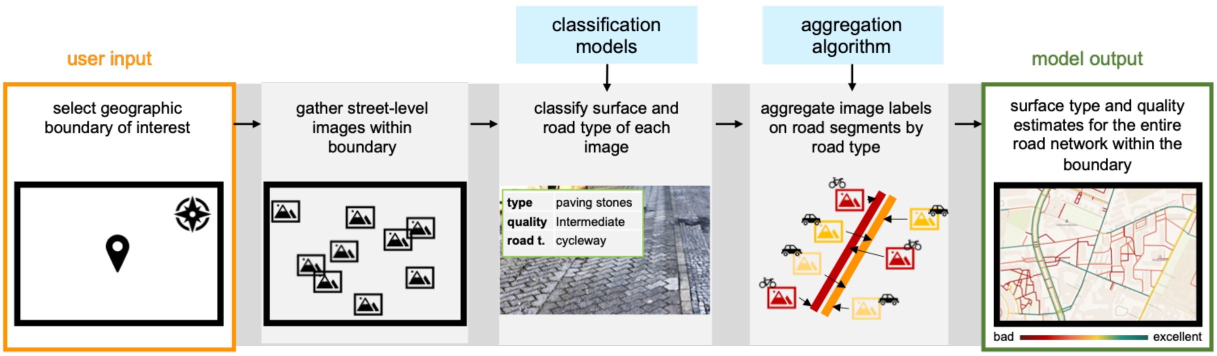


Figure 1: Schematic of the proposed pipeline from user input (geographic boundary) to model output (classified road network). Note that, for clarity, the diagram displays the aggregation algorithm for quality only.

Its smartphone app allows contributors to easily capture georeferenced image sequences during their trips by car, bicycle, or on foot, resulting in a diverse collection that includes roadways, cycleways, and footways. Additionally, Mapillary offers an API [4] for programmatic access, which we integrate into our software stack.

The accessibility and widespread use of Mapillary, along with the heterogeneity of the data regarding contributors, regions, devices, camera angles, and transportation modes, make it a suitable choice for training robust image classification models. Imagery from Mapillary, along with its metadata and data extracted using computer vision methods, has been employed for various use cases, including dataset creation for urban analytics covering a variety of traffic participants views [11, 18], generation of a cycleway network [8], and spatiotemporal analysis of city dynamics, e.g., in the context of walkability [26].

Other accessible data sources could be easily integrated into our pipeline, supplementing or even replacing Mapillary as the image database. Crowdsourcing platforms such as KartaView [2] and Panoramax [5], the latter currently used mainly in France, provide similar types of images, but have a substantially lower coverage in Germany. The commercial image platform Google Street View [1] offers panorama images of high quality due to the prevalence of standardized recording by the platform itself with professional equipment, but the data is not freely available. In addition, Google halted image publication in Germany for several years, only resuming in mid-2023. Moreover, unlike Mapillary, contribution by volunteers is possible only with advanced technology for recording panorama images.

2.2 Image classification models

We develop a supervised deep learning-based model that predicts, from a Mapillary image, (1) the *surface type* (e.g., 'asphalt') and (2) the *surface quality* (e.g., 'intermediate') which reflects the physical usability of a road segment for wheeled vehicles, particularly regarding its regularity or flatness. Specifically, we fine-tune EfficientNetV2-S [24], pre-trained on ImageNet, using the recently published dataset StreetSurfaceVis [13, 14] comprising 9,122 Mapillary images manually annotated by surface type and quality.

The labeling scheme closely aligns with that of OSM, with surface type values consisting of *asphalt*, *concrete*, *paving stones*, *sett*, and *unpaved*, while quality values range from *excellent* and *good* to *intermediate*, *bad* and *very bad*. For more details on the annotation process, refer to [14]. We train a classification model to predict the surface type, and one regression model per surface type predicting the surface quality.

According to evaluations from [14], for an 80:20 train-validation split and a test set comprising 776 images from geographically distinct cities, the type classification model performs well, achieving an accuracy (loss) of 0.96 (0.13) on the training data, 0.94 (0.19) on the validation data, and 0.91 on the test data. The F1 scores for individual type classes in the test data are all equal to or exceed 0.9, except for the 'concrete' surface type, resulting in a weighted average F1 score of 0.84. These results indicate that the model generalizes effectively across different locations. For the quality regression models, deviations from the true values are normally distributed and centered around 0, suggesting no systematic bias in the predictions. Regarding quality predictions, the overall Spearman correlation coefficient of 0.72 (ranging from .42 to .65 for individual type classes), an accuracy of 0.63, and a 1-off accuracy (considering neighboring classes as correct classifications) of almost 1.0 reflect a strong positive relationship between the predicted and true quality rankings. While the model effectively captures the relative ordering of surface quality, some variability remains, similar to human assessments, as quality classes are rather fluid [14].

Since roads, especially in urban settings, often consist of various sections such as roadways, cycleways, sidewalks, or other areas like parking lots or green stripes, we train an additional model to resolve this ambiguity. Specifically, we fine-tune a classifier, with an architecture similar to the surface type model, to distinguish between the following *road type* classes: *roadway*, *bike lane*, *cycleway*, *sidewalk*, *path*, and *no road or no single focus area*. The last category includes images that either lack a clear focus on any part of the road or mainly depict non-road elements, such as buildings or cars.

Based on 7,324 images with a train-validation split of 80:20, an initial model achieves an accuracy (loss) of 0.99 (0.03) for the training data and 0.88 (0.45) on the validation dataset. A weighted

F1 score of 0.88 is reached on the validation data, with class-specific F1 scores of 0.95 for *roadway*, 0.87 for *bike lane*, 0.76 for *cycleway*, 0.59 for *sidewalk*, 0.91 for *path*, and 0.82 for *no road or no single focus area* on the validation dataset.

2.3 Aggregation algorithm

To obtain a cohesive dataset for the entire road network within a geographic boundary, an algorithm aggregates classifications of individual images. We use the road segments provided in OSM as predefined aggregation units without further subdivisions, ensuring better compatibility with the utilized geographic database. Additionally, relying solely on class predictions can make it difficult to determine whether variations within a single road segment reflect actual differences in surface type or quality, which would justify a further splitting of a segment, or if they result from noise through factors such as imprecise geotags, incorrect classifications, or inconsistent surfaces.

Images within the region of interest are assigned to a road segment based on their geo-coordinates. If an image is near multiple road segments, such as a roadway and a cycleway (cf. image (b) in Figure 2), we use the predicted road type class for the respective images to resolve any ambiguity. Using all images successfully assigned to a road segment present in the underlying OSM network, the surface type is determined by majority vote, while the surface quality of the segment is calculated as an average.

Road segments may span multiple hundred meters, and certain parts may be overrepresented, such as when a certain driveway is highly frequented by one contributor. To prevent single parts from disproportionately influencing the overall road segment rating, we adjust the presented base idea by first computing aggregated values for 20-meter subsegments and then calculating unweighted aggregates of the values corresponding to these subsegments as the predicted value for the entire road segment. To remove unreliable results, a subsegment is required to have at least three agreeing type classifications, and a road segment to have classifications for at least half their subsegments, otherwise the value is considered missing.

It should be noted that, while OSM is a viable choice due to its high world-wide coverage and easy accessibility, other geographic databases can be substituted in the aggregation algorithm, particularly if they offer a more reliable or finer-grained network for the area of interest.

3 Evaluation in real-world scenarios

The pipeline development was informed by communication with German municipalities that could benefit from the enriched road network data. Specifically, we assessed the surface state on a provided **target cycleway network of Berlin**. Information on whether a cycleway is present on a road segment or if cyclists should use the roadway was provided for 80% of the segments. For the remaining 20%, this information was inferred based on road type classifications. Based on a manually labeled sample of this dataset, we obtained an accuracy of 0.91 and an average F1 score of 0.84 for surface type, along with a Spearman correlation of 0.53 and an accuracy of 0.62 for surface quality. These results suggest strong generalizability of the type model and a robust performance of the aggregation procedure. A more sensible aggregation algorithm in future work for surface quality could likely bear potential for performance improvements. Additionally, 20% of the road segments of the entire cycling network lacked classification values due to missing images. Thus, even in a large city like Berlin, where Mapillary has comparably very good coverage, certain parts of the infrastructure remain underrepresented.

(a) road type unclear from single image
surveyor99|123502363306902

(b) road with three road types: roadway, cycleway, and sidewalk
carlheinz|160748322585932

Figure 2: Examples of clear and ambiguous road types. Images from Mapillary; contributor names and image IDs are indicated for each image.

Municipalities and other interested parties can address these gaps to fully utilize the pipeline. One such example is a **rural municipality** in Germany who captured all their roads using a car-based setup and uploaded the images to Mapillary. As the images were captured consistently and in a high-quality homogeneous setup, issues related to image quality, recency, completeness, and road type classification were significantly reduced. However, while asphalt and unpaved roads were well distinguished, the type classification model struggled with paving stone surfaces. (Concrete and sett were excluded from this evaluation due to negligible occurrence in the dataset). This difficulty was likely caused by the different characteristics of rural roads compared to the predominantly urban settings in the training data, revealing limitations in transferability. Specifically, based on a manually labeled sample of 214 road segments, we obtained an accuracy of 0.76, with F1 scores of 0.94 for asphalt, 0.74 for unpaved, and 0.58 for paving stones. A Spearman correlation of 0.52 for surface quality and an accuracy of 0.66 align well with the evaluation of the Berlin use case.

4 Limitation and discussion

This paper proposes a pipeline that uses crowdsourced street-level imagery to classify road network surface type and quality. While initial evaluations in real-world scenarios demonstrated the feasibility of this approach, more extensive testing is required to assess the overall performance. The described pipeline constitutes an initial implementation, but several limitations and areas for improvement remain.

Firstly, the underlying image database, Mapillary, does not guarantee good coverage across all regions, with varying coverage of individual municipalities depending on contributor activity. Additionally, image quality may vary, the recency of images might not be provided, and images taken from a car perspective might be over-represented, potentially introducing bias toward certain road types. However, the infrastructure offered by the Mapillary app provides an easily accessible option for additional data collection, which can be utilized by communities, as demonstrated in our second example in Section 3.

Secondly, the current implementation relies on pre-defined road segments. However, road segments can exhibit non-uniform surface characteristics, and future iterations should include methods to subdivide segments when surface changes are detected in a robust and reliable manner.

Thirdly, the current aggregation mechanism for surface quality simply averages all values, without considering how different qualities may influence the overall rating. For example, if a subsegment is rated as 'bad', the entire road segment might need to obtain a low rating. As our initial evaluation for the Berlin cycleway network in Berlin in Section 3 suggests, further research into more meaningful aggregation may be beneficial.

Moreover, further research into road type classification could enhance accuracy and make it independent of underlying road network metadata. While current road scene segmentation models [21] fail to provide robust classifications on our training data, advancements in this field may offer superior results compared to image classification, as they provide more granular information. Alternatively, including additional training data sources, such as the newly released *Global Streetscapes* [11] street-view image dataset containing road type information, may improve model performance.

The generalizability of the classification models presents an additional challenge. While the training dataset StreetSurfaceVis is mindfully created to provide a heterogeneous set of images, the model will most likely not classify every potential road type equally well. A process should be implemented to extend the training data with cases the model struggles with, including, e.g., road types and surfaces prevalent in other countries.

Generally, transferability to other regions may pose an issue due to geographical limitation of the training data, the coverage of Mapillary images, and the coverage of OSM road network data, including metadata on road types. Additionally, our approach may encounter difficulties in regions experiencing rapid changes in their road network, as the images need to be temporally aligned.

References

[1] 2024. Google Street View. https://www.google.com/streetview/
[2] 2024. KartaView. https://kartaview.org
[3] 2024. Mapillary. https://www.mapillary.com/
[4] 2024. Mapillary API. https://www.mapillary.com/developer/api-documentation
[5] 2024. Panoramax. https://panoramax.fr
[6] Rishu Chhabra, C. Rama Krishna, and Seema Verma. 2021. A survey on state-of-the-art road surface monitoring techniques for intelligent transportation systems. *International Journal of Sensor Networks* 37, 2 (2021), 81–99. https://doi.org/10.1504/IJSNET.2021.118485
[7] PostGIS Project Steering Committee et al. 2018. PostGIS, spatial and geographic objects for postgreSQL. https://postgis.net
[8] Xuan Ding, Hongchao Fan, and Jianya Gong. 2021. Towards generating network of bikeways from Mapillary data. *Computers, Environment and Urban Systems* 88 (July 2021), 101632. https://doi.org/10.1016/j.compenvurbsys.2021.101632
[9] April Gadsby, James Tsai, and Kari Watkins. 2022. Understanding the Influence of Pavement Conditions on Cyclists' Perception of Safety and Comfort Using Surveys and Eye Tracking. *Transportation Research Record* 2676, 12 (Dec. 2022), 112–126. https://doi.org/10.1177/03611981221090936 Publisher: SAGE Publications Inc.
[10] P. M. Harikrishnan and Varun P. Gopi. 2017. Vehicle Vibration Signal Processing for Road Surface Monitoring. *IEEE Sensors Journal* 17, 16 (Aug. 2017), 5192–5197. https://doi.org/10.1109/JSEN.2017.2719865 Conference Name: IEEE Sensors Journal.
[11] Yujun Hou, Matias Quintana, Maxim Khomiakov, Winston Yap, Jiani Ouyang, Koichi Ito, Zeyu Wang, Tianhong Zhao, and Filip Biljecki. 2024. Global Streetscapes — A Comprehensive Dataset of 10 Million Street-Level Images across 688 Cities for Urban Science and Analytics. *ISPRS Journal of Photogrammetry and Remote Sensing* 215 (Sept. 2024), 216–238. https://doi.org/10.1016/j.isprsjprs.2024.06.023
[12] Jan Hrnčíř, Pavol Žileckỳ, Qing Song, and Michal Jakob. 2016. Practical multicriteria urban bicycle routing. *IEEE Transactions on Intelligent Transportation Systems* 18, 3 (2016), 493–504. https://doi.org/10.1109/TITS.2016.2577047 Publisher: IEEE.
[13] Alexandra Kapp, Edith Hoffmann, Esther Weigmann, and Helena Mihaljevic. 2024. *StreetSurfaceVis: a dataset of street-level imagery with annotations of road surface type and quality.* https://doi.org/10.5281/zenodo.11449977
[14] Alexandra Kapp, Edith Hoffmann, Esther Weigmann, and Helena Mihaljević. 2024. StreetSurfaceVis: a dataset of crowdsourced street-level imagery with semi-automated annotations of road surface type and quality. arXiv:2407.21454 [cs.CV]
[15] Ahmet-Serdar Karakaya, Leonard Thomas, Denis Koljada, and David Bermbach. 2023. A Crowdsensing Approach for Deriving Surface Quality of Cycling Infrastructure. arXiv:2305.01712
[16] Young-Mok Kim, Young-Gil Kim, Seung-Yong Son, Soo-Yeon Lim, Bong-Yeol Choi, and Doo-Hyun Choi. 2022. Review of Recent Automated Pothole-Detection Methods. *Applied Sciences* 12, 11 (Jan. 2022), 5320.
[17] Joseph Kurebwa, Tawanda Mushiri, and others. 2019. A study of damage patterns on passenger cars involved in road traffic accidents. *Journal of Robotics* 2019 (2019). Publisher: Hindawi.
[18] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. 2017. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* 4990–4999.
[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035.
[20] Jonathan Pearlman, Rory Cooper, Jonathan Duvall, and Ryan Livingston. 2013. Pedestrian Pathway Characteristics and Their Implications on Wheelchair Users. *Assistive Technology* 25, 4 (Oct. 2013), 230–239. https://doi.org/10.1080/10400435.2013.778915
[21] Lorenzo Porzi, Samuel Rota Bulò, Aleksander Colovic, and Peter Kontschieder. 2019. Seamless Scene Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*
[22] Thiago Rateke, Karla Aparecida Justen, and Aldo Von Wangenheim. 2019. Road Surface Classification with Images Captured From Low-cost Camera - Road Traversing Knowledge (RTK) Dataset. *Revista de Informática Teórica e Aplicada* 26, 3 (Nov. 2019), 50–64. https://doi.org/10.22456/2175-2745.91522
[23] Yashar Safyari, Masoud Mahdianpari, and Hodjat Shiri. 2024. A Review of Vision-Based Pothole Detection Methods Using Computer Vision and Machine Learning. *Sensors* 24, 17 (2024). https://www.mdpi.com/1424-8220/24/17/5652
[24] Mingxing Tan and Quoc Le. 2021. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning.* PMLR, 10096–10106.
[25] Oskar Wage, Udo Feuerhake, Christian Koetsier, Anne Ponick, Niklas Schild, Thido Beening, and Samsondeen Dare. 2020. Ride vibrations: Towards comfort-based bicycle navigation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 43 (2020), 367–373. https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-367-2020 Publisher: Copernicus GmbH.
[26] Meihui Wang, James Haworth, Huanfa Chen, Yunzhe Liu, and Zhengxiang Shi. 2024. Investigating the potential of crowdsourced street-level imagery in understanding the spatiotemporal dynamics of cities: A case study of walkability in Inner London. *Cities* 153 (2024), 105243. https://doi.org/10.1016/j.cities.2024.105243
[27] Wanli Ye, Wei Jiang, Zheng Tong, Dongdong Yuan, and Jingjing Xiao. 2021. Convolutional neural network for pothole detection in asphalt pavement. *Road Materials and Pavement Design* 22, 1 (Jan. 2021), 42–58. https://doi.org/10.1080/14680629.2019.1615533
[28] H Zakeri, Fereidoon Moghadas Nejad, and Ahmad Fahimifar. 2017. Image based techniques for crack detection, classification and quantification in asphalt pavement: a review. *Archives of Computational Methods in Engineering* 24 (2017), 935–977. https://doi.org/10.1007/s11831-016-9194-z Publisher: Springer.
[29] Qi Zhou, Zixian Liu, and Zesheng Huang. 2024. Mapping Road Surface Type of Kenya Using OpenStreetMap and High-resolution Google Satellite Imagery. *Scientific Data* 11, 1 (April 2024), 331. https://doi.org/10.1038/s41597-024-03158-7 Publisher: Nature Publishing Group.

MapYog – Intelligent Spatiotemporal Data Explorer

Ankit Sharma
YogLabs AI Research Foundation
Jaipur, India
ankit@yoglabs.ai

Madhur Gupta*
YogLabs AI Research Foundation
Jaipur, India
madhurg@yoglabs.ai

Mohd Junaid*
YogLabs AI Research Foundation
Jaipur, India
mohdj@yoglabs.ai

Anuj Nandwana*
YogLabs AI Research Foundation
Jaipur, India
anujn@yoglabs.ai

Lokendra P. S. Chauhan
Qen Labs Inc.
Minneapolis, USA
lokendra@qen.ai

Abstract

MapYog addresses the critical challenge of managing and analyzing heterogeneous, multi-granular geospatial data, a key issue in urban planning, environmental monitoring, and various geospatial applications. Existing systems often struggle to integrate diverse data types—actual, predicted, and simulated—across different spatial and temporal scales, limiting their capacity for in-depth, real-time analysis. MapYog overcomes these limitations through its innovative *geospatially and temporally coded (GT-coded) dataset abstraction model*[1], which unifies and curates data from multiple sources with customizable granularity. Unlike existing platforms, MapYog features a powerful UI built on deck.gl and kepler.gl, enabling sophisticated layer-based visualizations and dynamic auto-granularity zooming. Its decoupled microservice architecture further distinguishes it by offering enhanced scalability, flexibility, and modularity, separating preprocessing, model training, and API operations. These features make MapYog a more robust and adaptable solution for complex geospatial data challenges compared to traditional systems. **Application Release**[2]

CCS Concepts

• **Information systems** → **Geographic information systems**.

Keywords

UrbanAI, Interactive Data Visualization, Spatiotemporal Predictions

ACM Reference Format:
Ankit Sharma, Madhur Gupta, Mohd Junaid, Anuj Nandwana, and Lokendra P. S. Chauhan. 2024. MapYog – Intelligent Spatiotemporal Data Explorer. In *2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI (UrbanAI'24), October 29-November 1 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681780.3697250

*These authors conducted this work as employees during their internship at YogLabs.
[1]Across paper for brevity we refer GT-coded dataset (see Sec. 2.1) as simply 'dataset'
[2]The MapYog application is live, with an open-source release planned soon. Project details and source code will be available at: https://www.yoglabs.ai/mapyog/.

UrbanAI'24, October 29-November 1 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1156-5/24/10
https://doi.org/10.1145/3681780.3697250

1 Introduction

Globally, fine particulate matter air pollution caused 6.7 million premature deaths[3] and 93 billion cases[4] of illness in 2019. Urban heat islands, which trap heat absorbed by buildings, exacerbate these issues, increasing health costs and disease burdens. The interaction between urban heat and air pollution is a growing concern, and studying their combined effects is becoming more common in Urban AI research.

Multiple Data Sources In parallel with these challenges, there has been a proliferation of fine-grained hyperlocal urban data, driven by the widespread availability of advanced data collection technologies. The advent of Internet of Things (IoT) devices, satellite remote sensing, and other networked sensors has made it easier and more cost-effective to gather large volumes of real-time spatial data [1]. This *Actual data*, which includes air quality indices (AQI)[5][6], meteorological records[7][8], building heights[9], and even crowdsourced human sensing data [2][10], offers unprecedented opportunities for urban analysis and environmental monitoring.

Other valuable sources of data are predicted data from AI models and simulated data from AI-driven simulation tools. *Predicted data* provides insights through fine-scale forecasts, such as detecting wildfires or structural changes via AI-powered thermal remote sensing. *Simulated data*, generated by advanced models like GraphCast for weather forecasting, adds another layer of complexity.

Multiple Data Modalities Additionally, this geo data is multimodal, including time-series data from IoT sensors, remote sensing images, videos from surveillance cameras and drones, and tabular data from censuses and surveys. Most of it is geo-tagged, embedding spatial and temporal information crucial for urban analysis.

Multiple Data Granularities As urban planning evolves, the concept of the digital twin, originally from manufacturing and aerospace disciplines, has been extended to urban environments. Digital twins [2] enable the integration and simulation of multi-granular data, capturing varying levels of spatial and temporal detail. For instance, remote sensing data may provide broader regional insights, while drone imagery and surveillance cameras

[3]https://who.int/news-room/fact-sheets/detail/household-air-pollution-and-health
[4]https://internationalbanker.com/finance/air-pollution-is-responsible-for-critical-global-health-and-economic-problems
[5]https://www2.purpleair.com/
[6]https://www.epa.gov/data
[7]https://open-meteo.com/
[8]https://openweathermap.org/darksky-openweather-3
[9]https://www.overturemaps.org/
[10]https://cozie.app/

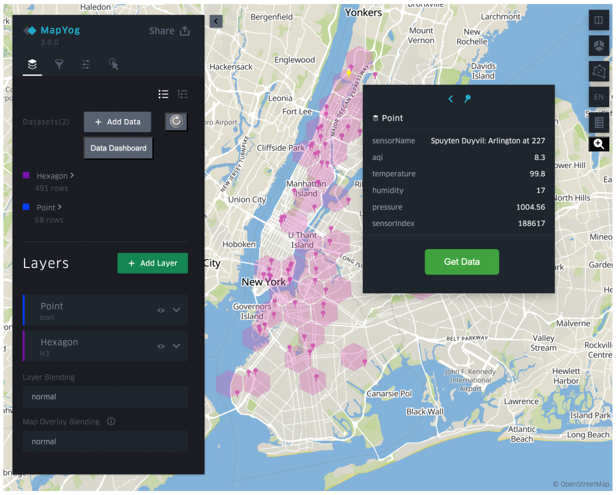


Figure 1: MapYog's Map User Interface (MapUI) showing point as well as area-level data based on the Air Quality Index (AQI) data aggregated by MapYog's Data Aggregator backend from Purple API (a sensor-based data collection API).

offer hyperlocal views. Similarly, temporal granularity ranges from hourly satellite images to second-level video data, creating a rich, layered understanding of urban dynamics.

Joint exploration of heterogeneous, multi-granular data in UrbanAI is essential for understanding complex urban environments. By examining relationships between different spatial scales and diverse data, insights emerge, such as the link between infrastructure changes and environmental shifts detected by thermal remote sensing. This joint approach benefits various stakeholders by improving quality of life, enabling advanced research, helping agencies monitor key factors, and allowing businesses to optimize resources – with visualization being crucial for interpreting this data.

Challenges Building a joint visual analytics system for heterogeneous, multi-granular urban data is challenging due to varied spatial scales and granularities. The system must offer flexible map visualization adaptable to different data types and scales, from small to large areas, and support both point- and area-level analysis. It should also integrate new data sources seamlessly, including IoT, satellite, crowdsourced, predictive, and simulated data. A decoupled design is essential for integrating data curation, analytics, and visualization, ensuring independent operation and easy updates.

Existing systems for environmental and urban data visualization largely fall into proprietary, paid services, such as Google's environment APIs[11] and India's Ambee[12], which focus primarily on visualization rather than data curation. While these services provide estimates for various environmental metrics like air quality and pollens, they often do not address the complexities of integrating and curating diverse data sources. Notable visualization-focused projects include Heigit[13], Deck.gl[14] for extensive layer-based map visualizations, and Cesium[15] for 3D maps. Advances in the Digital Twin development community have improved spatial data visualization, but current applications often neglect the spatial accuracy and reliability of urban digital twin components [3].

Recent efforts have seen some progress in integrating data and visualization, with libraries like Uber's Kepler.gl[16] providing more automated and flexible visualization tools. However, integration of machine learning and simulation models into these systems remains limited, with few exceptions like the NUS's Microclimate Digital Platform[17], which has restricted open-source access. Additionally, data curation capabilities are notably absent from many existing tools, such as Kepler.gl, which offers limited data upload functionality. Overall, there is a clear gap in end-to-end integrated systems that combine robust data curation with complex data visualization, providing easy access and usability for stakeholders dealing with heterogeneous and multi-granular data.

To address these gaps, here are the key aspects and contributions of the **MapYog system**:

- **Powerful UI Interface**: MapYog offers a powerful UI with a Map UI built on deck.gl and kepler.gl for advanced layer-based visualization of multi-granular data, and a Data UI for listing, loading, and curating datasets with customizable spatial and temporal granularity.
- **Dataset Abstraction**: MapYog's ***GT-coded dataset abstraction*** streamlines curation of heterogeneous, multi-granular data into uniform spatiotemporally constrained tables or objects, enabling flexible management of diverse data types, metrics, spans, and granularities.
- **Data Curation and Integration**: MapYog excels in managing data collection and model-based predictions, integrating and aggregating diverse data sources through backend job scheduling with customizable spatio-temporal granularities.
- **Decoupled Architecture**: MapYog's decoupled architecture uses dataset abstraction for consistent data management and API-driven microservices to separate preprocessing and model training, ensuring flexibility and modularity in data handling and analytics.
- **Advanced Visualization Features**: It offers dynamic auto-granularity zooming with H3 spatial indexing, supports custom visualization layers, and enables interactive temporal data visualization for both real-time and historical data.
- **Data Analytics Capabilities**: MapYog provides spatial and spatio-temporal filtering, layer-based analysis, and joint visualization of multiple data layers for comprehensive insights.

2 Architecture Overview

The MapYog system architecture (see Figure 2) is divided into frontend (yellow) and backend (red) components, each with specialized modules that interact seamlessly. The frontend includes the MapUI for interactive geospatial data visualization, and the Data UI for dataset selection and curation. The backend, built on Flask, handles requests from the frontend and includes the Data Support module, which manages dataset initialization, temporal data fetching, and scheduling data aggregation tasks. The Data Aggregator module,

[11]https://mapsplatform.google.com/maps-products/#environment-section
[12]https://www.getambee.com/
[13]https://heigit.org/projects/
[14]https://deck.gl/
[15]https://cesium.com/platform/cesiumjs/
[16]https://kepler.gl/
[17]https://cde.nus.edu.sg/arch-ucdl/researches/ucdl-microclimate-digital-platform/

composed of the Data Collector, Predictor, and Simulator, is responsible for retrieving, predicting, and simulating data. The database stores all system data, while the Tileserver provides map styling and base map customization.

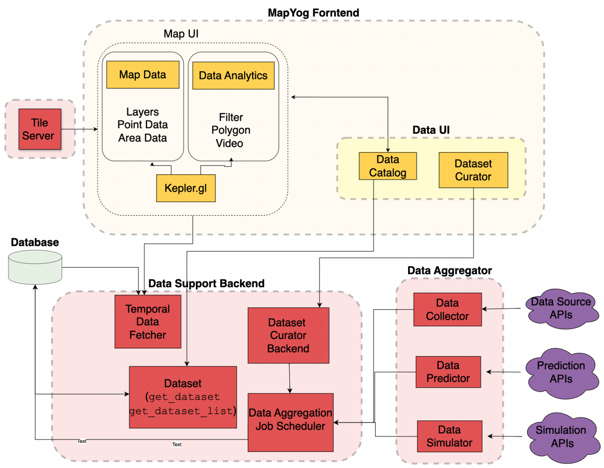
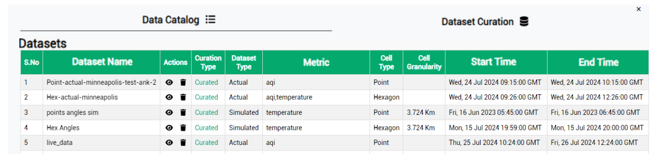
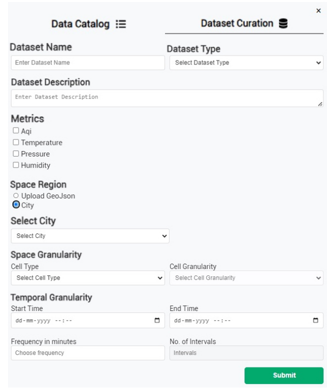
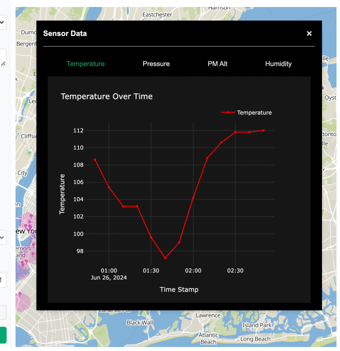


Figure 2: High Level Architecture of MapYog



(a) Dataset Catalog UI





(b) Dataset Curation UI (c) On-click temporal graph pop-up

Figure 3: MapYog's Data & Temporal UI

We first describe the *dataset* abstraction, which plays a key role in MapYog. Following that, we discuss the dataset curation workflow within the architecture. Afterward, we will cover the Map-based Data Visualization, and lastly, delve into the Data Analytics.

2.1 Dataset Abstraction

At the heart of MapYog is its '*GT-coded dataset*' abstraction, streamlining the curation of diverse, multi-granular data from both actual and model-based sources. Its key properties include:

Span: The '*span*' attribute defines the spatio-temporal window within which data is measured or estimated. Temporal spans can include past, future, or ongoing periods, categorizing datasets into *historical* or *live* categories. Spatial spans define the geographical area of interest, such as states, cities, or counties, enabling localized data analysis.

Granularity: A significant focus of MapYog is on granularity, which is subdivided into temporal and spatial dimensions. Temporal granularity breaks down the time span into finer intervals, allowing for detailed time-based analysis, while spatial granularity divides the spatial span into point-level or area-level data, accommodating different scales of geographic data.

This dataset abstraction differs from typical machine learning datasets because each data point is bound by geospatial and temporal constraints, determined by the dataset's *span* and *granularity* properties. For instance, a point-level dataset would have each data point linked to specific geo-coordinates within its spatial span, such as AQI sensor readings in New York. Similarly, an area-level dataset, like a grid, would associate data points with square cells based on the dataset's granularity, such as 1 km x 1 km cell-level estimates from an AI-based AQI model in New York. Additionally, the data points are temporally coded, meaning they correspond to particular moments or intervals within a temporal span. We define such dataset as a *geospatially and temporally coded (GT-coded) dataset.*

Metric: Each dataset revolves around one or more metrics, such as AQI or temperature, representing the quantity measured or estimated at a specific spatiotemporal span and granularity. A metric is a meaningful value that can be associated with a point or area at a particular span and granularity i.e. it can be GT-coded. For instance, a raw remote sensing image may not provide immediate insights, but extracting land usage within building boundaries from that image transforms it into a meaningful metric.

Source Type: Datasets in MapYog can come from a range of sources, including actual data from sensors and satellites, predictions from AI models, or simulated data from simulations. A valid source must eventually provide GT-coded metric data, allowing for the seamless integration of various data types.

It's important to note that source data may require transformation into GT-coded data. For example, surveillance camera footage isn't GT-coded by default, but extracting a relevant metric—such as the pedestrians counts on a geospatially defined sidewalk, structured using a grid like H3 indexing—makes it so. Similarly, predictive or simulation models should produce GT-coded outputs, either by directly generating GT-coded data such as, predicting AQI estimates at a granularity like 10m x 10m, or through post-processing.

2.2 Dataset-Driven Data Curation

MapYog employs a dataset-driven approach to data curation, where the Data UI (see Figure: 3 (a,b)) acts as the primary interface, facilitating interaction with the Data Curation backend and the database. This process begins with dataset initialization, which creates a database table linked for the dataset and stores its meta-information.

Actual Data Collection: When curating actual data, a data aggregation job is scheduled via the Data Aggregation Job Scheduler (see Figure 2). This job is responsible for collecting the selected metric's data (e.g., AQI, temperature) from the data source API for a specified spatiotemporal span and granularity. The Data Collector submodule of the Data Aggregator module fetches the necessary data, which is then stored in the initialized dataset, effectively connecting the dataset with the data source API.

Integration with ML & Simulation Models: For data derived from machine learning (ML) models or simulations, a similar job scheduling approach is used. The job aggregates predictions from the selected metric's prediction or simulation API, utilizing recent seed data to generate forecasts. The job supplies this seed data from live datasets, facilitating real-time predictions or simulations. For instance, live AQI data can be used to forecast future conditions, with the results stored in a separate, pre-initialized dataset. This method also supports historical data predictions, allowing for granularity adjustments, such as converting point-level sensor data into area-level estimates like county-wide AQI values.

The curated datasets can be viewed and loaded using the Dataset Catalog (see Figure 3 (a)) which is also part of Data UI.

2.3 Decoupled Architecture

MapYog's architecture is designed with a strong emphasis on decoupling, achieved through a dataset-driven communication protocol and API-driven microservices. The dataset-driven abstraction standardizes the integration of heterogeneous, multi-granular data—whether historical, live, predicted, or collected—by storing all data types as datasets. This uniform approach ensures seamless communication between jobs and various source, prediction, or simulation APIs. Additionally, the architecture further decouples processes by keeping raw data preprocessing behind source APIs and model training and deployment behind prediction APIs, allowing for modular and independent system components.

2.4 GIS Database and Custom Data Upload

MapYog's backend is powered by a cloud-hosted GIS database, where datasets are stored as spatiotemporally coded objects. Each dataset is defined by its specific granularity, geocoding, and temporal span, ensuring precise spatial and temporal data organization. The database is decoupled from the core system via an API, aligning with the dataset abstraction and allowing flexibility in database selection, including specialized GIS or spatio-temporal databases. Additionally, MapYog supports the self-upload of datasets, enabling users to explore and curate custom data.

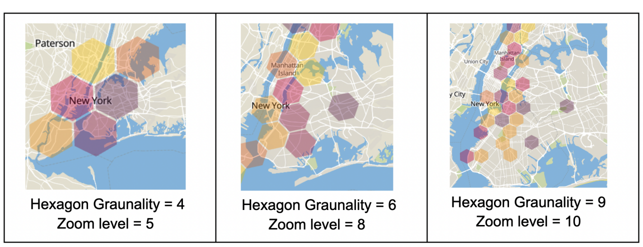


Figure 4: Granularity based automatic zooming functionality

2.5 Map-Based Data Visualization and Analytics in MapYog

MapYog leverages the kepler.gl library, which builds upon deck.gl, to provide advanced map-based data visualization and analytics. By using layered visualization, MapYog allows users to overlay multiple variables on a single map, facilitating comprehensive data exploration. Key features include auto-granularity zooming (see Figure 4), which employs H3-based[18] spatial indexing to adjust visual detail dynamically based on zoom level. Additionally, users can apply parameter-based filtration, toggle layer visibility, and engage with 3D visualizations, enhancing the clarity of spatial data.

Custom visualization layers are a significant aspect of MapYog's capabilities, supporting specific aggregation, filtering, and visualization needs across 2D, 3D, and temporal dimensions. The platform extends kepler.gl's point data pop-ups to include area-level information for various granularities and integrates temporal data visualization (see Figure 3 (c)). This includes real-time and historical data displays through interactive pop-ups based on dataset status. MapYog's custom tile server, using OpenStreetMap Tiles, provides a flexible and configurable base map style.

In terms of data analytics, MapYog inherits kepler.gl's robust analytical features, including spatial-filtering such as polygon-based filters, spatio-temporal video processing, and layer-based data filtering. The platform enables joint analysis by filtering and blending multiple layers, adjusting colors to visually combine overlapping data effectively.

3 Conclusion and Future Directions for MapYog

MapYog has made notable progress in geospatial data management with its GT-coded dataset abstraction model, enabling efficient handling of multi-granularity and heterogeneous data. By integrating actual, predicted, and simulated data, it supports detailed spatial and temporal analytics. Its decoupled microservice architecture boosts scalability, making it a powerful tool for complex data challenges.

MapYog's future development will focus on enhancing data analytics, database capabilities, and data management. It plans to improve temporal analysis with multi-metric time series and correlation studies, and enable more complex analyses like comparative analysis across spatial and temporal spans and inter-granularity variations.

The database will be upgraded by experimenting with distributed systems like Apache Sedona to improve scalability and efficiently manage multi-granularity data.

MapYog will introduce live data visualization for real-time streams and simulated historical data, along with user profiling and dataset ownership for better access control. It will expand to diverse domains like surveillance and health records, addressing data granularity challenges to enhance adaptability and robustness.

References

[1] Mihai Andronie, George Lăzăroiu, Mariana Iatagan, Iulian Hurloiu, Roxana Ștefănescu, Adrian Dijmărescu, and Irina Dijmărescu. 2023. Big data management algorithms, deep learning-based object detection technologies, and geospatial simulation and sensor fusion tools in the internet of robotic things. *ISPRS International Journal of Geo-Information* 12, 2 (2023), 35.

[2] Fabian Dembski, Uwe Wössner, Mike Letzgus, Michael Ruddat, and Claudia Yamu. 2020. Urban digital twins for smart cities and citizens: The case study of Herrenberg, Germany. *Sustainability* 12, 6 (2020), 2307.

[3] Soheil Sabri, Kostas Alexandridis, Mehrdad Koohikamali, Sonya Zhang, and H Erkan Ozkaya. 2023. Designing a Spatially-explicit Urban Digital Twin Framework for Smart Water Infrastructure and Flood Management. In *2023 IEEE 3rd Intern. Conference on Digital Twins and Parallel Intelligence (DTPI)*. IEEE, 1–9.

[18] https://h3geo.org/

Optimization of Site Selection for Free-Floating Shared Electric Vehicles Based on Deep Reinforcement Learning

Shaohua Wang†
Aerospace Information Research Institute
Chinese Academy of Sciences
Beijing 10094, China
wangshaohua@aircas.ac.cn

Xiaojian Liang
Faculty of Geomatics
Lanzhou Jiaotong University
Lanzhou 730070, China
105803638@qq.com

Liang Zhou
Faculty of Geomatics
Lanzhou Jiaotong University
Lanzhou 730070, China
zhougeo@126.com

Xiao Li
Faculty of Geomatics
Lanzhou Jiaotong University
Lanzhou 730070, China
11220869@stu.lzjtu.cn

Yongyi Pan
Faculty of Geomatics
Lanzhou Jiaotong University
Lanzhou 730070, China
3542248892@qq.com

Yin Cheng
School of Architecture and Urban Planning
Lanzhou Jiaotong University
Lanzhou 730070, China
cyin0511@qq.com

Chunxiang Cao
Aerospace Information Research Institute
Chinese Academy of Sciences
Beijing 100094, China
caocx@aircas.ac.cn

Cheng Su
Aerospace Information Research Institute
Chinese Academy of Sciences
Beijing 100094, China
sucheng23@mails.ucas.ac.cn

Jiayi Zheng
Aerospace Information Research Institute
Chinese Academy of Sciences
Beijing 100094, China
zhengjiayi@aircas.ac.cn

ABSTRACT

As a contemporary mode of transportation for medium-to-long distances, the widespread adoption of free-floating shared electric vehicles has the potential to reduce urban carbon emissions and alleviate traffic congestion. However, this mode of transportation also faces challenges such as irregular parking and supply-demand imbalances. This study proposes a spatial optimization method for the deployment of free-floating shared electric vehicles. The research employs data sources such as population, points of interest, and public parking lots in Shenzhen, utilizing spatial analysis methods to explore the spatial distribution relationship between public parking lots and influencing factors. By combining spatial distribution characteristics, a maximum coverage location model is established, and a deep reinforcement learning algorithm is utilized to solve the model, yielding optimized results. The efficacy of traditional algorithms is compared with that of the deep reinforcement learning algorithm, demonstrating the latters superiority in solving location models. This study presents a novel approach to optimizing the configuration of shared transportation planning in Shenzhen, offering a valuable reference for the location selection of other facilities.

†Corresponding author

UrbanAI'24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1156-5/24/10.
https://doi.org/10.1145/3681780.3697249

CCS CONCEPTS

• **Computing methodologies** → *Deep reinforcement learning*

KEYWORDS

Free-Floating Shared Electric Vehicles, Spatial Analysis, Deep Reinforcement Learning, Maximal Coverage Location Problem

ACM Reference Format:

Shaohua Wang, Xiaojian Liang. et al. 2024. Optimization site selection of free-floating shared electric vehicles combined with deep reinforcement learning. In *2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI (UrbanAI24), October 29 -November 1, 2024, Atlanta, GA, USA*. ACM, New York, NY, USA, 7 pages.
https://doi.org/10.1145/3681780.3697249

1 INTRODUCTION

As environmental awareness continues to grow, an increasing number of people are inclined to reduce their environmental impact through sharing methods. The sharing economy, an emerging economic model [1], optimizes resource use through sharing, leasing, or borrowing, which reduces waste and pollution while promoting sustainable urban development [2]. Shared mobility, an important field within the sharing economy, enables flexible and free travel through shared means, such as shared cars, bicycles, and scooters. Shared Electric Vehicles (SEVs), used for medium and long-distance travel [3], offer convenience and flexibility. They enable users to travel short distances within cities and take longer trips without owning a car, thereby reducing travel and maintenance costs. The widespread adoption of SEVs can reduce fuel consumption, minimize resource waste, and alleviate urban traffic congestion.

The current research on SEVs is primarily concerned with the location of charging stations [4], the optimization of rental points [5], and the planning of relocation and pricing strategies for SEVs [6]. However, limited research has focused on the location optimization of Free-Floating Shared Electric Vehicles (FFSEVs). As a novel, flexible, and convenient mode of sharing, existing research predominantly emphasizes balancing vehicle downtime with the number of charging stations [7]. This is achieved through the use of mixed-integer linear programming combined with demand forecasts based on national mobility and taxi trip data, which are employed to plan operational areas [8]. Wang et al. [9] have proposed an algorithm based on deep reinforcement learning, which uses attention mechanisms and deep reinforcement learning to solve the Maximum Coverage Location Problem (MCLP) [10]. This algorithm has been demonstrated to provide efficient and rapid solutions for MCLP across various problem scales, offering a new perspective for subsequent spatial optimization problems [11]. Zhong et al. [12] have developed a new method, called ReCovNet, which leverages deep reinforcement learning to address the Maximum Coverage Billboard Location Problem (MCBLP). This method achieves a good balance between efficiency and accuracy and has been successfully applied to a real case in New York City.

To address current research gaps in optimizing FFSEV deployment areas using artificial intelligence, this study combines SEV spatial distribution characteristics, establishes a maximum coverage model, and applies a deep reinforcement learning algorithm for optimal allocation. By comparing with traditional algorithms, this study demonstrates that deep reinforcement learning methods can effectively guide SEV optimization.

2 DATA AND METHODS

2.1 Study Area

Shenzhen is a modern city in China that has been significantly affected by the reform and opening-up policy. It has experienced a period of rapid economic development and is one of China's Special Economic Zones, which serves as an international comprehensive transportation hub. The population of Shenzhen is highly mobile, particularly in the city center and commercial areas such as Futian District and Luohu District. This places significant demands on the transportation system. The majority of commuters rely on the metro and bus systems, yet congestion during peak hours persists as a significant challenge that requires immediate attention. The optimization of the allocation of shared electric vehicles can significantly alleviate traffic pressure and serve as a model for shared transportation planning in similar cities.

2.2 Data

The operational model of free-floating shared electric vehicles is contingent upon the distribution of parking lots. Consequently, this study employs public parking lot data from Shenzhen to simulate the deployment areas of free-floating shared electric vehicles. The comprehensive data used in this study includes population data, public parking space data, road network data, and Points of Interest (POI) data. The public parking lot data was sourced from the Amap Open Platform as of May 30, 2024. The dataset contains facility names, facility addresses, latitude and longitude coordinates, and public parking lot IDs, totaling 5,992 records. Population data was sourced from the WorldPop website with a resolution of 100 meters. Shenzhen's road network data was obtained from GLOBIO's GRIP global road database.

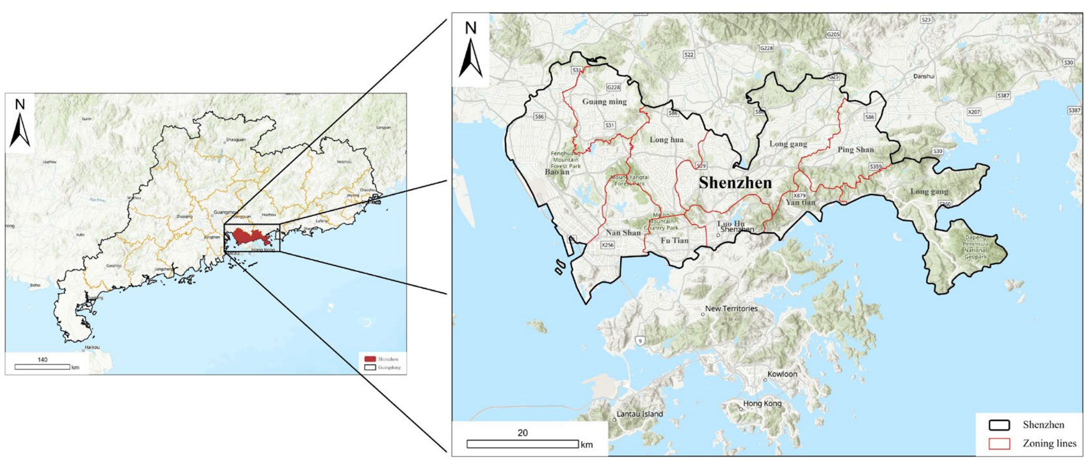


Figure 1: Study Area.

POIs are typically employed to describe specific locations or places on a map that are attractive or of practical value to people. The analysis of POI points facilitates the comprehension of a regions economic development, tourism resources, transportation facilities, and other pertinent aspects. Through research and analysis, it was found that POI data for residential areas, schools, shopping malls, accommodation services, leisure and entertainment, transportation facilities, attractions, and industrial parks have a significant correlation with the distribution of parking lots. This finding aids in further exploring the usage patterns of free-floating shared electric vehicles.

All these data points are sourced from Amap. A spatial analysis of the parking lot dataset revealed that the majority of public parking lots are concentrated in the Shenzhen-Hong Kong Innovation Cooperation Zone in Futian District, the Exhibition and Ocean City in Baoan District, the Comprehensive Innovation Core Zone jointly formed by Baoan and Nanshan Districts, and the Guangming Science City in Guangming District. The remaining parking lots are distributed in the high-tech industrial zones of various districts. As shown in Figure 2, These areas are all situated in the city center and are distinguished by their highly developed economies, convenient urban transportation, and dense populations. Consequently, this study conducts a more comprehensive investigation into the factors influencing the development of public parking lots, with a particular focus on the aforementioned commonalities.

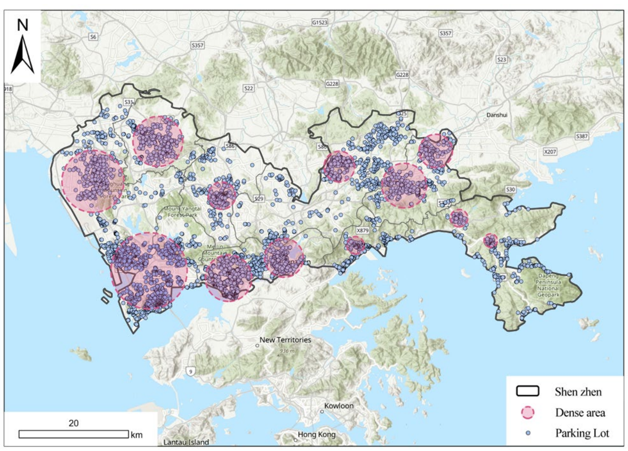


Figure 2: Spatial distribution of public parking lots in Shenzhen.

2.3 Methods

2.3.1 Maximum Coverage Location Problem Model. This study employs a Maximal Coverage Location Problem (MCLP) model, which aims to cover the largest possible demand area using a limited number of facility points. The maximal coverage model can be represented as follows:

$$E = Maximize \sum_{i \epsilon N} Si\, Qi \tag{1}$$

$$s.t. \qquad S_i \leq \sum_{j=1}^{M} a_{ij}\, T_j, \qquad \forall i = 1,2,\dots,N \tag{2}$$

$$\sum_{j=1}^{M} T_j \leq P \tag{3}$$

$$T_j \in \{0,1\}, \qquad \forall j = 1,2,\dots,M \tag{4}$$

$$S_i \in \{0,1\}, \qquad \forall i = 1,2,\dots,N \tag{5}$$

where E represents the maximum amount of covered demand; i denotes the demand points, j and denotes the facility points; N and M are the sets of demand points and facility points, respectively; P is the number of facility points allowed to be established; R is the service radius of a facility point; Q_i represents the demand at demand point i; T_j indicates the selection status of facility point j, with T_j =1 if facility point j is selected, and T_j =0 otherwise; S_i indicates the coverage status of demand point i, with S_i =1 if demand point i is covered, and S_i =0 otherwise; a_{ij} represents the coverage of demand point i by facility j, if facility j covers demand point i, then a_{ij}=1, otherwise a_{ij}=0; d_{ij} represents the distance from demand point i to facility point j.

2.3.2 Deep Reinforcement Learning. Deep reinforcement learning is a hybrid of reinforcement learning and deep learning that enables an agent to interact continuously with its environment to obtain the optimal solution strategy. The agent perceives the state of the environment and learns to select an appropriate action based on the received reward feedback to maximize long-term benefits. The environment receives a series of actions from the agent and evaluates them, providing feedback to the agent. This cycle continues until convergence or until the predefined number of training iterations is reached, as shown in Figure 3. Consequently, it exhibits considerable efficacy and extensive potential for deployment in addressing intricate decision-making challenges.

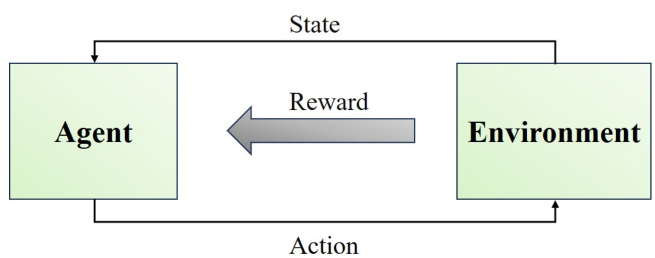


Figure 3: Deep Reinforcement Learning Framework.

2.3.3 Workflow. Due to the constraints imposed by the limitations in equipment conditions and data availability, this study selects Futian District in Shenzhen, which is characterized by a high population density, high mobility, and a dense transportation network, as the study area. The district is divided into 200×200 grid cells, with the center point of each grid cell designated as the demand point for the area. In order to evaluate the demand of a region, it is necessary to consider the influence of its surrounding environment in a comprehensive manner.

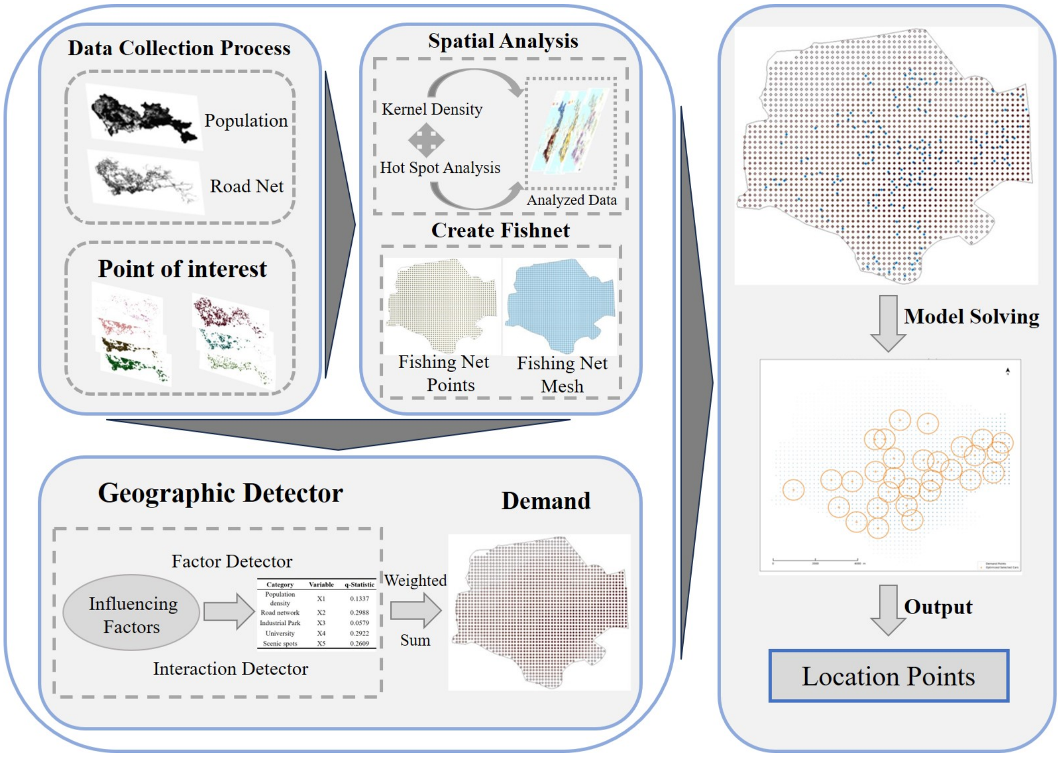


Figure 4: The workflow for optimizing the location of FFSEV.

As the deployment area of free-floating shared electric vehicles includes public parking lots, this study employs kernel density analysis and hot spot analysis to explore the spatial relationships between population, transportation networks, POIs, and public parking lots. Geographic detectors are employed to calculate the influence weight of each factor on public parking lots.

First, kernel density analysis is conducted for each influencing factor, and the values are extracted to the demand points. Subsequently, geographic detectors are employed to calculate the influence weight of each factor, and a weighted sum is performed to obtain the demand for each demand point. Finally, a maximal coverage model is established, and deep reinforcement learning is employed to solve the model, resulting in the final optimized results. The results are then compared with the efficiency and coverage of traditional algorithms. Figure 4 shows the workflow for optimizing the location of the FFSEV.

3 Experiments and Analysis

3.1 Influencing Factors

3.1.1 Transportation Network. The distribution of the transportation network has a direct impact on the smoothness of traffic flow and the selection of travel routes. The density of the transportation network directly affects the convenience of travel, thereby influencing the distribution of parking demand. The application of hot spot analysis to the road network in Shenzhen, as illustrated in Figure 5, reveals the presence of hot spot areas distributed across the eastern portion of Bao'an District, the southeastern coastal areas, and extending to the central Longgang District. These areas are connected to Nanshan District and Futian District. Other notable hotspot areas are located in Longhua District. The cold spot areas are located in Dapeng New District, Pingshan District, the southwestern part of Longgang District, the northwestern part of Guangming District, and the northern part of Bao'an District. The areas with a high density of parking lots exhibit a significant overlap with the hot spot areas of road density.

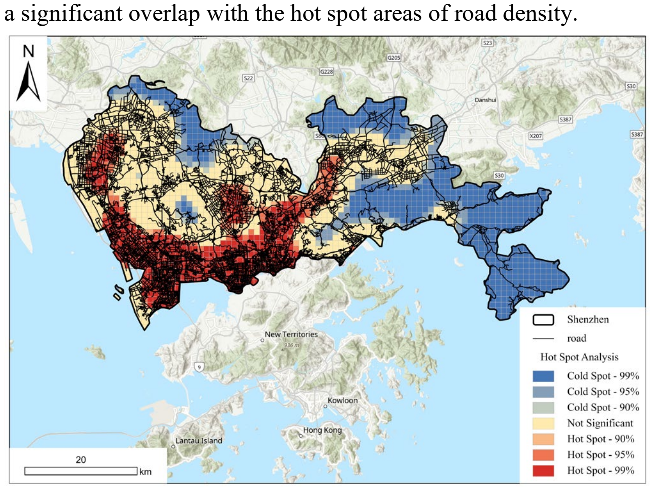


Figure 5: Hot spot analysis of the transportation network.

3.1.2 Population Density. Population density is a fundamental factor in spatial analysis, as it reflects the distribution of people within a given area. A high population density is typically indicative of a greater prevalence of social activities and economic exchanges, accompanied by a correspondingly greater consumer demand. This study employs a spatial overlay of the distribution of parking lots with population density to reflect their relationship. Areas with high population kernel density values exhibit a high degree of correlation with those with high parking lot values. As shown in Figure 6, The aforementioned areas are primarily situated in Futian District, the southwestern portion of Luohu District, the central-north region of Bao'an District, and the comprehensive innovation core zone, which is formed by the southern parts of Bao'an District and Nanshan District.
These regions are home to a significant number of high-tech and innovative enterprises, as well as numerous high-end residential communities. They also offer a wide range of living and entertainment facilities, and boast a rich cultural and educational infrastructure. Consequently, these areas experience a considerable increase in traffic loads and commuter volumes. The introduction of free-floating shared electric vehicles has the potential to enhance vehicle mobility, reduce the number of privately owned vehicles on the road, and consequently, reduce traffic congestion.

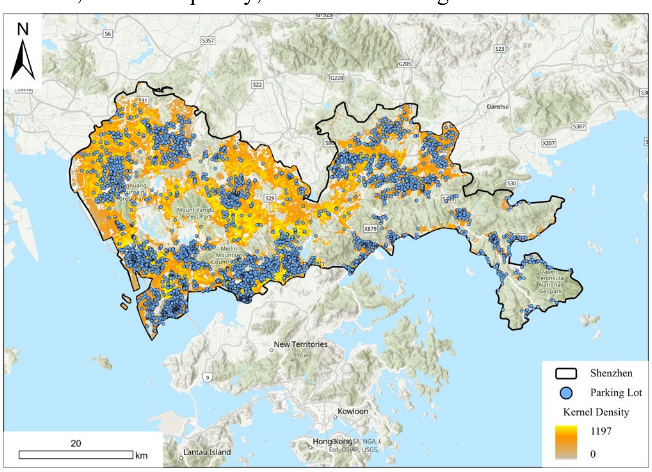


Figure 6: Population density and parking lot distribution.

3.2 Geographic Detector

The primary functions of the Geographic Detector include factor detection, interaction detection, risk detection, and ecological detection [13]. This study employed factor detection and interaction detection to assess the explanatory power of influencing factors on the dependent variable, as well as the interaction effects among different factors on the dependent variable. The quantity of public parking lots was considered the dependent variable, while the numerical values of various influencing factors were treated as independent variables to explore their explanatory power and significance regarding the quantity of public parking lots. The specific values of the dependent and independent variables, along with the results, are presented in Table 1.

Table 1: Results of Factor Detector.

Category	Variable	q-Statistic	p-Value
Population density	X1	0.1337	0.00
Road network	X2	0.2988	0.00
Industrial Park	X3	0.0579	0.00
University	X4	0.2922	0.00
Scenic spots	X5	0.2609	0.00
Mall	X6	0.6456	0.00
Recreational sports	X7	0.5080	0.00
Residential area	X8	0.5057	0.00
Accommodation services	X9	0.4648	0.00
Transportation facilities	X10	0.4727	0.00

As shown in Figure 7, the interaction detector indicates a general enhancement among all factors. This indicates that the distribution of parking lots is influenced by the interaction of multiple factors, rather than being solely controlled by a single factor. In particular, there is a notable correlation between the university (X4) and the residential area (X8), the latter of which is also linked to the mall (X6) and transportation (X10). Additionally, there is a significant relationship between the scenic spot (X5) and the mall (X6). Conversely, the interaction between population density (X1) and industrial park (X3), as well as between population density (X1) and road network (X2), is relatively weak. A pronounced positive influence is evident among the remaining factors.

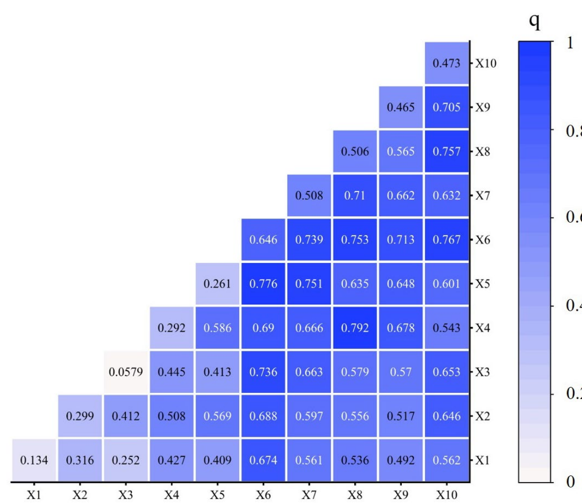


Figure 7: The results of the interaction detector.

3.3 Location Optimization

Due to the limitations of the available equipment and the size of the model, the demand points are selected from the center of the 200× 200m grid points in Futian District, which totals 1,797 demand points. For the purpose of identifying potential sites, the Arcpy software is employed to filter the data, with public parking lots within a 200-meter radius classified as large-scale public parking lots, resulting in 159 facility points. In this study, we employ deep reinforcement learning to address the MCLP model, identifying 30 deployment areas for free-floating shared electric vehicles with a service radius of 500 meters each [14]. The data is imported and divided into two sets: demand points and facility points. A total of 12,800 training samples and 2,000 validation samples are generated automatically. The epoch size is set to 12,800, and the batch size is set to 640. Upon completion of 277 iterations of the training process, the difference in the loss function reaches a minimum and the algorithm begins to converge. This ultimately satisfies the condition of maximum demand.

Furthermore, this study compares the solution time, coverage volume, and coverage rate of traditional algorithms, namely genetic algorithm (GA) and Gurobi solver, with deep reinforcement learning (DRL), as illustrated in Figure 8 and Table 2. It can be observed that Gurobi achieves the best coverage volume and coverage rate, thereby providing the optimal solution for the maximum coverage model. Although the coverage rate of DRL is slightly lower than that of Gurobi, it is higher than that of GA. With regard to solution time, the time required by Gurobi is 37 times that of DRL, while the time required by GA is 7.7 times that of DRL. This suggests that DRL is more efficient and produces superior solutions in the context of site optimization.

(a) Input Data (b) GA

(c) DRL (d) Gurobi

Figure 8: Results of different solving methods.

Table 2: Different Solving and Results.

Solution method	Solution time(s)	Objective value	Coverage rate(%)
GA	2.038	258	42.36
Gurobi	9.820	273	44.83
DRL	0.265	271	44.50

4 CONCLUSION

This study proposes an optimization method for configuring the Maximum Coverage Location Model using deep learning algorithms to address issues such as parking difficulties and supply-demand imbalances of FFSEV. Taking Shenzhen as an example, this study explores the spatial distribution characteristics of free-floating shared electric vehicles through the spatial distribution characteristics of public parking lots. A variety of factors influencing the distribution of public parking lots are analyzed, revealing a high degree of correlation between areas of high population density and dense transportation networks with public parking lots, particularly in the Futian District of Shenzhen. The Geographic Detector is employed to elucidate the explanatory power of each factor on parking lot distribution. Shopping centers exhibit the highest explanatory power, while industrial parks exhibit the lowest. The weights corresponding to each factor are then derived. By weighting and summing, the demand for each demand point is determined. Subsequently, a Maximum Coverage Model is established for the Futian District, Shenzhen, using deep reinforcement learning algorithms to obtain optimization results. The efficacy of deep reinforcement learning in solving site optimization models is validated by a comparison with traditional algorithms.

Nevertheless, it should be noted that this study is not without its limitations. Due to the limitations of the model size and the data constraints, the use of public parking lots to simulate the spatial distribution of FFSEVs may introduce some errors. Furthermore, the limited capacity of the device precludes the selection of extensive research areas. Future work will continue to optimize the model, improve solution efficiency, and increase model size in order to more accurately address real-world problems.

ACKNOWLEDGMENTS

This study was financially supported by Talent Introduction Program Youth Project of the Chinese Academy of Sciences (E2Z10501), innovation group project of the Key Laboratory of Remote Sensing and Digital Earth Chinese Academy of Sciences (E33D0201-5), Henan Zhongmu County Research Project (E3C1050101), CBAS project 2023, Remote Sensing Big Data Analytics Project (E3E2051401).

REFERENCES

[1] Thomas Puschmann and Rainer Alt. 2016. Sharing economy. *Business & Information Systems Engineering* 58 (2016), 93-99.

[2] Marc Saleh, Alexandre Milovanoff, I Daniel Posen, Heather L MacLean, and Marianne Hatzopoulou. 2022. Energy and greenhouse gas implications of shared automated electric vehicles. *Transportation Research Part D: Transport and Environment* 105 (2022), 103233.

[3] Riku Viri and Johanna Mäkinen. 2024. EV-share development: speed vs interest to adopt. *European Transport Research Review* 16, 1 (2024), 10.

[4] Georg Brandstätter, Michael Kahr, and Markus Leitner. 2017. Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand. *Transportation Research Part B: Methodological* 104 (2017), 17-35.

[5] Ömer Kaya, Kadir D Alemdar, Ahmet Atalay, Muhammed Y Çodur, and Ahmet Tortum. 2022. Electric car sharing stations site selection from the perspective of sustainability: A GIS-based multi-criteria decision making approach. *Sustainable Energy Technologies and Assessments* 52 (2022), 102026.

[6] Ulrik Eilertsen, Olav M Falck-Pedersen, Jone V Henriksen, Kjetil Fagerholt, and Giovanni Pantuso. 2024. Joint relocation and pricing in electric car-sharing systems. *European Journal of Operational Research* 315, 2 (2024), 553-566.

[7] Mohammad S Roni, Zonggen Yi, and John G. 2019. Optimal charging management and infrastructure planning for free-floating shared electric vehicles. *Transportation Research Part D: Transport and Environment* 76 (2019), 155-175.

[8] Lukas Felician Krasel, Georg Brandstätter, and Bin Hu. 2023. Area of Operation Planning for Free-Flating Car Sharing Systems. *Applied Sciences* 13, 14 (2023), 8408.

[9] Shaohua Wang, Haojian Liang, and Yang Zhong. 2023. DeepMCLP: Solving the MCLP with Deep Reinforcement Learning for Urban Spatial. (2023).

[10] Haojian Liang, Shaohua Wang, Huilai Li, Liang Zhou, Hechang Chen, Xueyan Zhang, and Xu Chen. 2024. Sponet: solve spatial optimization problem using deep reinforcement learning for urban spatial decision analysis. *International Journal of Digital Earth* 17, 1 (2024), 2299211.

[11] Ershun Zhong. 2022. Deep Mapping—A Critical Engagement of Cartography with Neuroscience. *Geomatics and Information Science of Wuhan University* 47, 12 (2022), 1988-2002.

[12] Yang Zhong, Shaohua Wang, Haojian Liang, Zhenbo Wang, Xueyan Zhang, Xi Chen, and Cheng Su. 2024. ReCovNet: Reinforcement learning with covering information for solving maximal coverage billboards location problem. *International Journal of Applied Earth Observation and Geoinformation* 128 (2024), 103710.

[13] Zehua Zhang, Yongze Song, and Peng Wu. 2022. Robust geographical detector. *International Journal of Applied Earth Observation and Geoinformation* 109 (2022) 102782.

[14] Simone Weikl, and Klaus Bogenberger. 2015. A practice-ready relocation model for free-floating carsharing systems with electric vehicles–Mesoscopic approach and field trial results. *Transportation Research Part C: Emerging Technologies* 57 (2015) 206-223.