Naqsha-e-Safar

Identifying & Classifying Public Transit Deserts in Lahore Using Graph Neural Networks

Information Technology University
Department of Computer Science

Mujtaba Asim	BSCS22012@itu.edu.pk
Muhammad Rehman	BSCS22018@itu.edu.pk
Aleena Basit	BSCS22005@itu.edu.pk
Abdullah Raffay	BSCS22085@itu.edu.pk
Eman Afrooz	BSCS22099@itu.edu.pk

December, 2025

Abstract

Public transportation accessibility is crucial for urban equity and sustainable development. This research tackles the problem of identifying transit deserts in Lahore—areas with high population density but poor access to public transport. We built a spatial graph of the city using population data from WorldPop, transit routes from CityLines and Zameen, and the road network from OpenStreetMap. By applying Graph Neural Networks, specifically a Graph Convolutional Network, we classified around 160,000 points of interest into transit-accessible and transit-desert regions. Our approach partitions the city into roughly 160 communities using the Louvain algorithm and uses a weak supervision strategy to generate training labels from obvious examples. The trained GCN identifies underserved areas that should be prioritized for future transport planning. While our results have some limitations due to incomplete data, they represent a solid first step toward data-driven transit equity analysis in Lahore, with the methodology being easily scalable to other cities.

Introduction

Background and Motivation

Urban transportation systems are basically the lifeline of modern cities—they enable economic activity, social connections, and access to essential services like healthcare and education. But the thing is, public transit infrastructure isn't distributed equally across cities. This creates what we call "transit deserts": areas where people have limited or no access to reliable public transport despite needing it. Lahore, being Pakistan's second-largest city with over 13 million people, faces this exact problem. Even with recent investments like the Orange Line Metro Train and Speedo bus routes, large portions of the city remain underserved.

Transit deserts aren't just about distance to the nearest stop. They're about the intersection of high population density, limited transportation options, and the resulting social and economic exclusion. People living in these areas end up spending more time and money on commutes, have reduced access to job opportunities, and generally face lower quality of life. For urban planners trying to allocate resources fairly, pinpointing these areas with accuracy is essential.

Traditional methods for measuring transit accessibility—like simple distance buffers or isochrone analysis

—don’t really capture the complexity of how cities work. They miss things like network topology, population distribution patterns, and the spatial relationships between different neighborhoods. That’s where our approach comes in.

Research Objectives

The goal of this project is to develop a data-driven method for identifying transit deserts in Lahore using Graph Neural Networks. Specifically, we aim to:

1. **Build a spatial graph**: Model Lahore as a graph where nodes represent points of interest and edges represent road connections, integrating population, transit, and infrastructure data.
2. **Engineer meaningful features**: Create node features that capture demographic characteristics, topological properties, and geographic context.
3. **Detect communities**: Use the Louvain algorithm to partition the city into spatial communities that reflect natural clustering in the urban fabric.
4. **Classify nodes**: Train a GNN to distinguish between transit-adequate and transit-deficient areas using a weak supervision approach.
5. **Generate actionable insights**: Produce visualizations and metrics that urban planners can actually use to guide transit expansion decisions.

Contributions

Our work makes a few key contributions:

- **Novel application**: This is one of the first attempts at using GNNs for transit desert identification in a developing country context, where data is often messy and incomplete.
- **Scalable framework**: The methodology relies entirely on publicly available datasets and is modular enough to be adapted to other cities without major changes.
- **Weak supervision strategy**: We introduce a systematic way to generate training labels from “obvious” examples, which lets us do semi-supervised learning without expensive manual labeling.
- **Practical validation**: We demonstrate that the approach works on real-world data from Lahore and produces results that align with intuition about underserved areas.

Related Work

Transit Accessibility Analysis

Most traditional approaches to measuring transit accessibility use distance-based metrics, like the classic “400m walking distance” buffer around stops, or isochrone maps that show areas reachable within a certain time. Gravity models have also been popular, where accessibility is weighted by the attractiveness of destinations. While these methods are straightforward and interpretable, they struggle to account for the complex interactions between population distribution, multi-modal transit options, and the actual topology of road networks. Recent work has started incorporating GIS-based techniques to get more nuanced assessments, but most of it still relies on fairly simple spatial analysis.

Graph Neural Networks in Urban Computing

GNNs have become really popular for modeling relational data in cities. They’ve been applied to traffic prediction, where the road network is naturally a graph, as well as air quality forecasting and urban land use classification. The key advantage of GNNs is their ability to aggregate information from neighboring nodes through message-passing, which captures the spatial autocorrelation that’s everywhere in urban data. Despite this, their use for transit equity and accessibility analysis is still pretty limited, especially in cities outside the developed world where the challenges are different.

Community Detection in Urban Graphs

Community detection algorithms are used to find clusters of densely connected nodes in graphs. The Louvain method is one of the most popular because it’s fast and works well on large networks. It’s been applied to urban graphs before—for example, to identify neighborhoods based on mobility patterns or to understand the functional structure of cities. Label propagation and DBSCAN are other common approaches. These techniques give us a natural way to group fine-grained spatial data into regions that make sense for planning purposes.

Methodology

Data Collection and Preprocessing

We pulled together data from multiple public sources, each providing a different view of Lahore's urban environment.

Population Density Data

Population estimates came from the WorldPop project, which provides gridded population data at 1km × 1km resolution. This data is derived from census records, satellite imagery, and geospatial modeling. For Lahore, it captures both the dense urban core and the lower-density periphery pretty well.

The tricky part was integrating this raster data with our graph representation. Initially, we tried a simple approach: for each point of interest, assign it the population value from the nearest grid cell. But this turned out to be a bad idea—it artificially inflated population density based on how many nodes happened to fall near a particular grid cell. If you had ten POIs clustered near one population reading, they'd all get assigned the same value, making it seem like the area had way more people than it actually did.

We fixed this by distributing each population reading evenly among all nodes within a 1km radius (matching the span of each grid cell). So if a cell has 10,000 people and there are 50 nodes within 1km of its center, each node gets credited with 200 people. This gives a much more realistic representation of population distribution across the graph.

Transit Infrastructure Data

Transit route data came from two main sources:

- **CityLines**: An open-source database of transit systems worldwide. We used it to get the geometry of Lahore's Orange Line Metro Train route.
- **Zameen**: A local property portal that published information about Speedo bus routes. We manually cleaned this data, which involved geocoding addresses, validating stop locations, and fixing inconsistencies.

In total, we have 519 stops from the metro system and 70 unique Speedo stops, giving us 589 transit stops across the city. We're aware this is incomplete —it doesn't capture informal transport like qingqis, private vans, or intercity buses that also serve neighborhoods—but it's the best publicly available data we could find.

Road Network and Points of Interest

We used OSMnx to download Lahore's road network and points of interest from OpenStreetMap. OSM has pretty good coverage of Lahore thanks to local mappers. We filtered POIs to include places relevant to daily life: shops, schools, hospitals, government offices, parks, etc.

This gave us roughly 140,000 nodes and 200,000 edges, forming the structural backbone of our urban graph.

Graph Construction

The city is modeled as a spatial graph $G = (V, E)$ where nodes V are points of interest and edges E represent physical road connections.

Node Insertion and Connectivity

We inserted the 589 transit stops as additional nodes into the graph. To make sure these weren't isolated, we connected each stop to its nearest existing POI. This is important for community detection—isolated nodes would mess up the algorithm's ability to identify connected regions. It also reflects the reality that transit stops are access points connected to the broader urban fabric.

Feature Engineering

Each node $v \in V$ gets a feature vector that captures different aspects of its urban context:

Geometric Features: Latitude and longitude (x, y) encode spatial position.

Demographic Features: Population density p_v is assigned using the distribution method described earlier —each node gets a fraction of the population from nearby 1km grid cells.

Topological Features: Street count (node degree) $\deg(v)$ indicates how connected a location is via the road network.

Accessibility Features: Distance to the nearest transit stop d_v is computed using network distance (actual walking distance along roads) rather than Euclidean distance.

Infrastructure Features: Binary flag $t_v \in \{0, 1\}$ indicating whether the node itself is a transit stop.

Community Features: One-hot encoded community membership derived from the Louvain algorithm (explained next).

Community Detection

We used the Louvain method to partition the graph into communities. This algorithm optimizes modularity—a measure of how much more densely connected nodes are within communities compared to between communities. It works in two phases that repeat iteratively: first, each node is assigned to the community that gives the biggest modularity increase; second, the graph is coarsened by treating each community as a single node, and the process repeats.

For our graph, Louvain identified around 160 communities. These range from compact clusters in the dense city center to more scattered groupings on the periphery. The communities don't necessarily match official administrative boundaries or residential societies, but they do reflect natural clusters in the urban network.

We tried a couple of alternatives. Label propagation worked but generated way too many communities (500+), making subsequent analysis unwieldy. We also looked at DBSCAN, but the communities it produced were ambiguous—probably due to parameter tuning issues on our end. We didn't want to waste too much time debugging it, so we stuck with Louvain. In hindsight, DBSCAN might have been useful for identifying linear transit corridors rather than compact regions, which could be interesting for future work.

Label Generation Strategy

One of the main challenges with this project was the lack of ground-truth labels. Manually labeling 160,000 nodes would be completely impractical. We solved this using weak supervision: automatically generate labels for "obvious" examples and treat everything else as unlabeled test data.

Defining Transit Deserts

A transit desert is characterized by high population density (lots of people who need transit) combined with poor accessibility (far from stops). Conversely, transit-adequate areas either have low population (less need for service) or good accessibility (need is being met).

Labeling Criteria

Let μ_p be the median population density across all nodes. We assigned labels $y_v \in \{0, 1\}$ as follows:

$$y_v = \begin{cases} 1 \text{ if } p_v > \mu_p \text{ and } d_v < 1500\text{m} \\ 0 \text{ if } p_v < \mu_p \text{ and } d_v > 4500\text{m} \\ \text{unlabeled otherwise} \end{cases}$$

Nodes with $y_v = 1$ are transit-adequate: either they're high-population areas with nearby stops, or low-population areas where transit isn't as critical. Nodes with $y_v = 0$ are used as negative examples to provide contrast—low-population areas far from stops. The "unlabeled" category includes everything in between: areas where it's not immediately obvious whether they're adequately served or not.

This gives us roughly 15-20% of nodes as labeled training data and 80-85% as the test set. The model has to learn from the clear cases and generalize to the ambiguous ones.

Graph Neural Network Architecture

We used a Graph Convolutional Network (GCN) for node classification. GCNs are well-suited for this because they naturally aggregate information from neighboring nodes, which is exactly what we want —transit accessibility has spatial extent, so knowing about nearby nodes is informative.

Our architecture has two GCN layers with 64 and 32 hidden units respectively, followed by a softmax output layer for binary classification. We used ReLU activations and applied dropout (rate=0.3) to prevent overfitting.

The model was trained for 200 epochs with early stopping based on validation loss. Training took about 15 minutes on a standard laptop CPU, which is pretty reasonable given the graph size.

Community-Level Aggregation

After getting node-level predictions from the GCN, we aggregated results to the community level. For each community C_i, we computed the transit adequacy ratio:

$$r_i = \frac{|\{v \in C_i : \hat{y}_v = 1\}|}{|C_i|}$$

Communities with low r_i values (high proportion of nodes predicted as transit-inadequate) are flagged as priority areas for transit investment. This aggregation smooths out noise from individual predictions and provides insights at a scale that's actually useful for planning—nobody's going to build transit based on individual POI-level data.

Results

Model Performance

The GCN successfully classified the test set. We don't have ground-truth labels for validation, so traditional accuracy metrics aren't really meaningful here. But the predictions show strong spatial coherence—neighboring nodes tend to get similar classifications, which is what we'd expect if the model is learning real patterns rather than just noise.

Identified Transit Deserts

The analysis highlighted several types of underserved areas:

Peripheral Communities: Neighborhoods along the city's eastern and southern edges, particularly beyond the Ring Road, show very poor transit access. These are areas with growing residential development but minimal formal transit coverage. People living here basically have to rely on expensive private transport or informal options.

Industrial Zones: Some industrial areas, especially those without dedicated worker transport, came up as transit deserts. These have high daytime populations from commuters but limited public transit options.

Interstitial Neighborhoods: There are pockets within the urban core that fall between major transit corridors. They're not directly served by metro or Speedo routes, and while they might be "close" to transit by Euclidean distance, the actual walking distance through the road network is significant.

Spatial Patterns

Looking at the classified communities on a map, a few clear patterns emerge:

- Transit adequacy is concentrated along the Orange Line corridor and major Speedo routes.
- There's a sharp gradient moving away from these corridors—accessibility drops off quickly beyond 2-3 km.
- Point of interest density is higher in the city center, which probably biases the GNN's attention toward well-mapped central areas.
- Peripheral communities are more spatially dispersed, reflecting lower OSM mapping coverage. This might underestimate the actual population and need for transit in those areas.

Limitations and Caveats

There are several important limitations to keep in mind:

Incomplete Transit Data: We only have the Orange Line and partial Speedo coverage. We're missing informal transport entirely—qingqis, private vans, intercity buses. Our accessibility estimates are definitely a lower bound; many "deserts" probably have some informal service.

Community Misalignment: The Louvain communities don't match administrative boundaries or residential societies. This makes it harder to directly apply results to policy decisions, which usually operate at the level of official districts or neighborhoods.

OSM Coverage Bias: OpenStreetMap coverage is better in wealthier, central areas. This creates an artificial density gradient in our POI distribution, and the GNN might be picking up on this rather than actual transit need.

Static Analysis: We're looking at a snapshot of current conditions. We don't model temporal aspects like rush-hour frequency, crowding, or planned infrastructure expansions.

Despite these issues, we think the results are a reasonable first approximation and demonstrate that the GNN approach can work for this problem.

Discussion

Methodological Insights

Using GNNs for transit desert identification has some clear advantages over traditional approaches. The ability to incorporate different types of features —demographic, topological, geographic—in a single framework is powerful. And the message-passing mechanism naturally captures the fact that transit accessibility isn't just about the nearest stop; it has spatial extent that affects surrounding areas.

The weak supervision strategy worked pretty well, though it does make assumptions about what counts as "obvious." The thresholds we chose (1500m and 4500m for distance, median population for density) are based on common urban planning heuristics, but different thresholds would produce different training sets. It would be good to do sensitivity analysis to see how robust the results are.

Computational Considerations

The project had some workflow challenges worth mentioning. We initially worked in Jupyter notebooks, which caused problems—memory issues during geospatial operations, state management bugs, version control conflicts with notebook metadata. For anything beyond quick prototyping, proper scripts with explicit memory management are the way to go.

Coordinate Reference System issues were another pain point. We wasted hours debugging CRS mismatches between datasets. The lesson: convert everything to a consistent projected CRS (EPSG:32643 for Pakistan) upfront, and only use geographic CRS (EPSG:4326) for final visualization.

Alternative Approaches

We explored some other modeling strategies that didn't make it into the final version:

Graph Attention Networks: We initially wanted to use GATs, which learn attention weights over edges. This could better capture that different road connections have different importance. But training was prohibitively slow—over 1.5 hours per epoch. We think the large number of one-hot encoded community features was the problem. If we used manual community partitioning instead, reducing feature dimensionality, GATs might be feasible and could perform better.

Graph Classification: Instead of classifying individual nodes, we considered treating each community as its own subgraph and doing graph-level classification. This would require manually labeling a smaller number of communities but might give more stable predictions and better align with planning decisions.

DBSCAN for Communities: As mentioned earlier, we tried DBSCAN but it didn't work well with our default parameters. With more tuning, it might be useful for identifying linear transit corridors or other non-compact structures that Louvain misses.

Practical Implications

For people actually doing urban planning in Lahore, our results suggest a few concrete directions:

Priority Corridors: The peripheral transit deserts should be high priority for new bus routes or BRT corridors. Radial routes connecting outer neighborhoods to the city center would have the biggest impact.

Last-Mile Connectivity: Even near existing transit, poor walkability—missing sidewalks, unsafe crossings, physical barriers—can create effective deserts. Investing in pedestrian infrastructure around stops could expand their coverage area without building new routes.

Demand-Responsive Transit: In dispersed peripheral communities, fixed-route service might not be efficient. Flexible options like micro-transit or ride-sharing integration could be more cost-effective.

Transit-Oriented Development: Zoning decisions should consider transit access. Encouraging dense, mixed-use development along existing and planned corridors maximizes the return on infrastructure investment.

Scalability and Generalization

One of the best things about this approach is how easy it is to adapt to other cities. The entire pipeline uses public datasets—WorldPop, OpenStreetMap, and whatever transit data is available locally. With minor tweaks to handle different road network structures and transit systems, you could run the same analysis on any city.

The modular design helps too. Data preprocessing, graph construction, community detection, and classification are all separate stages. You can swap out components—use a different community detection algorithm, try a different GNN architecture—without rebuilding everything from scratch.

Conclusion

This project shows that Graph Neural Networks are a viable tool for identifying transit deserts in Lahore. By integrating population, infrastructure, and network data into a spatial graph, we were able to classify areas by transit adequacy and flag underserved regions for priority action.

The results, while preliminary and subject to data limitations, provide a useful starting point for evidence-based transit planning. The methodology is scalable and could be applied to other Pakistani cities or adapted for international use with minimal changes.

The main takeaway is that data-driven approaches to transit equity are feasible even in contexts where data is messy and incomplete. With continued improvement in open datasets and refinement of the methods, tools like this could play a real role in making cities more accessible and equitable.

Future Work

Several directions could extend and improve this work:

Data Enhancement: The biggest limitation is incomplete transit data. Incorporating informal transport networks would give a much more complete picture of actual accessibility. Crowdsourced data collection, maybe through a mobile app, could help fill these gaps. Real-time data on transit frequency and reliability would also be valuable—a stop that's "nearby" but only served once every two hours isn't that useful.

Temporal Dynamics: Right now, the analysis is static. Extending it to consider time-varying accessibility—peak vs. off-peak service, time-of-day patterns —would better reflect commuters' lived experiences. We could also model how accessibility changes as new routes come online or as populations shift.

Equity Dimensions: Overlaying socioeconomic data would let us assess how transit deserts disproportionately affect vulnerable populations: low-income residents, elderly people, persons with disabilities. This would strengthen the equity focus and help prioritize interventions that reduce inequality.

Validation: We don't have ground-truth labels, which makes it hard to definitively validate the results. Household travel surveys, focus groups, or comparisons with existing accessibility studies could provide empirical confirmation. Even just checking whether our predictions match planners' intuitions would be useful.

Route Optimization: Right now, we identify underserved areas but don't suggest solutions. The next step would be route optimization: given the transit deserts, where should new routes or stops be placed to maximize coverage and ridership? This is a harder problem but directly actionable.

Multi-City Comparison: Applying the same methodology to multiple cities—Karachi, Islamabad, Rawalpindi—would enable comparative analysis. We could see which cities have worse transit equity, identify best practices, and understand how different urban forms and policies affect accessibility.

Alternative Models: With more computational resources or manual community curation, we could revisit GATs or try other GNN architectures. Graph transformers are another emerging approach that might work well.

Integration with Planning Tools: Ultimately, this should be integrated into the tools planners actually use. Building a web interface where planners can upload data, run the analysis, and visualize results would make it much more practical for real-world use.

The broader vision is to move transit planning away from ad-hoc decisions and toward systematic, data-driven approaches that explicitly center equity and accessibility. This project is one small step in that direction.