Naqsha-e-Safar

BSCS22005/12/18/85/99

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

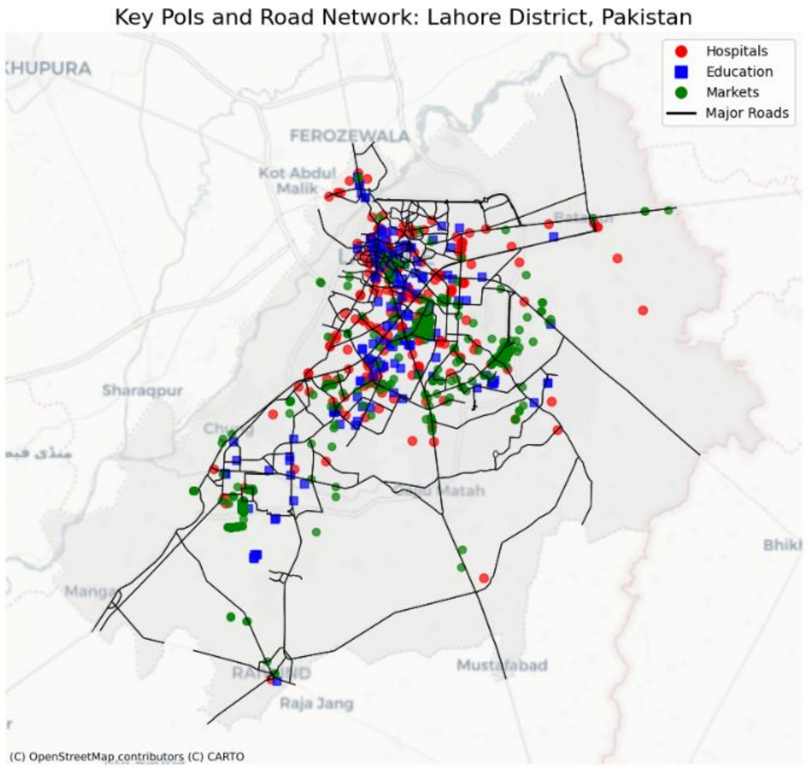
This project identifies areas in Lahore with high population density but poor access to public transport i.e “transit deserts.” Using population data (WorldPop), transit routes (CityLines), and the city’s road network (OpenStreetMap), we model Lahore as a spatial graph and apply a Graph Neural Network to classify underserved regions. The outcome is a data-driven map of transit accessibility gaps to guide future transport planning.

Datasets

1. [WorldPop (Population Density Data)] (https://hub.worldpop.org/geodata/summary?id=48110)
2. [OpenStreetMap] (https://www.openstreetmap.org/)
3. [CityLines] (https://www.citylines.co/lahore)
4. [Zameen - Speedo Bus Routes in Lahore] (https://www.zameen.com/blog/speedo-bus-routes-lahore.html)

Steps (01)

- Programmatically download and clean up data
- Manually cleaned up Speedo’s CSV data
- Get Lahore’s points of interest (PoI) via OSMNX
- Feature engineering: PoI geometry

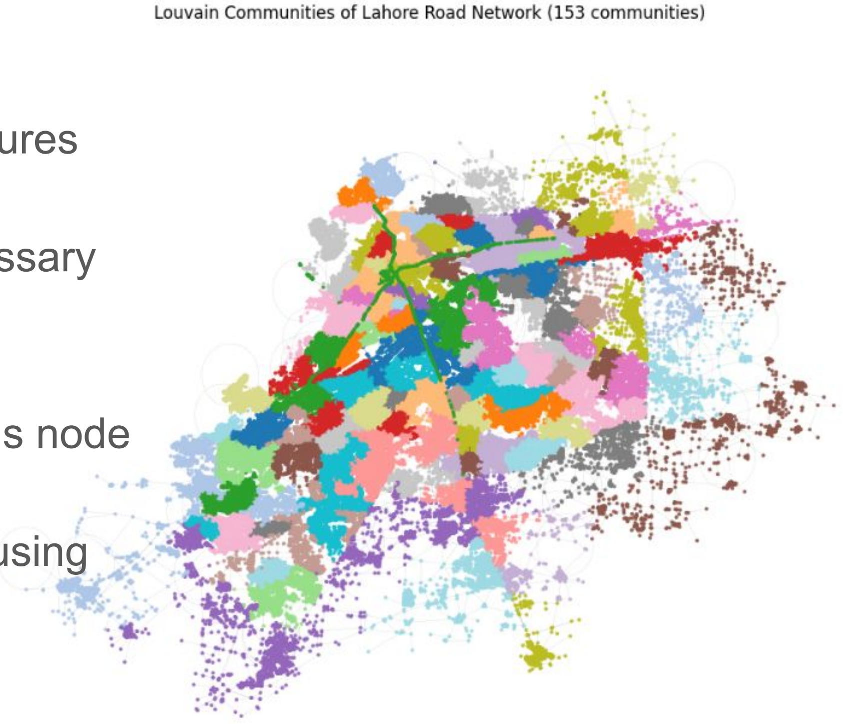

Key PoIs and Road Network: Lahore District, Pakistan
Hospitals
Education
Markets
Major Roads
(C) OpenStreetMap contributors (C) CARTO

Steps (02)

- Load Metro + Speedo data into separate GeoDataFrames
- Insert new nodes into graph
  - To ensure connectivity, we add an edge between a stop and the nearest other PoI (necessary for community detection)
  - Nodes: ~140,000 // Edges: ~200,000
- Add features: distribute population evenly amongst all nodes in vicinity & compute distance to nearest public transit stop
  - These were deemed as crucial features for the GNN
  - We were previously quantizing every PoI to its nearest population reading, but that inflated population density erroneously

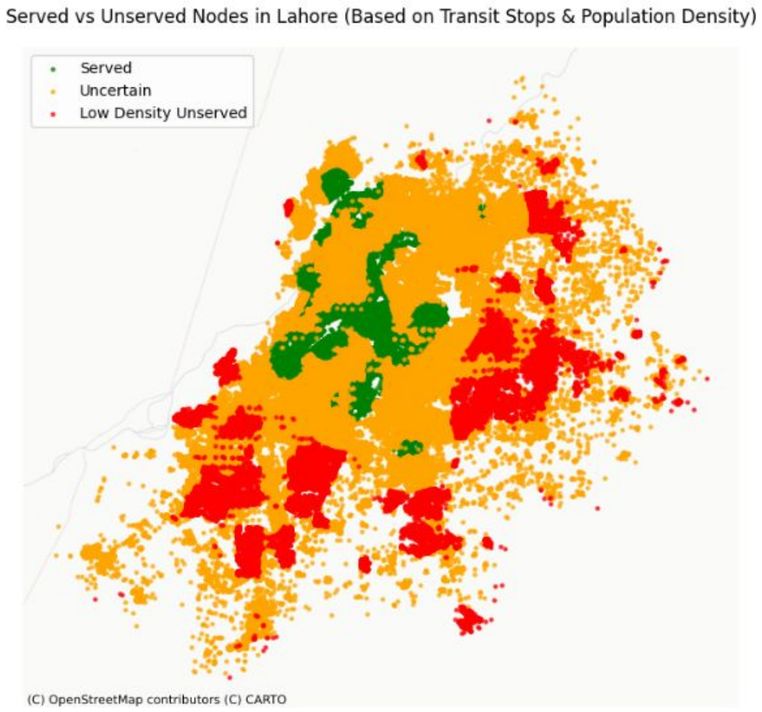
Steps (03)

- Run Louvain to partition ~160 communities
- One-hot encode communities as node features
- Feature Management
  - Convert to numeric types where necessary
  - Remove string attributes
  - Generate labels
    - Label corresponds to whether this node is in a transit desert or not
    - Idea: pick very ‘obvious’ nodes, using tight constraints


Louvain Communities of Lahore Road Network (153 communities)

Steps (04)

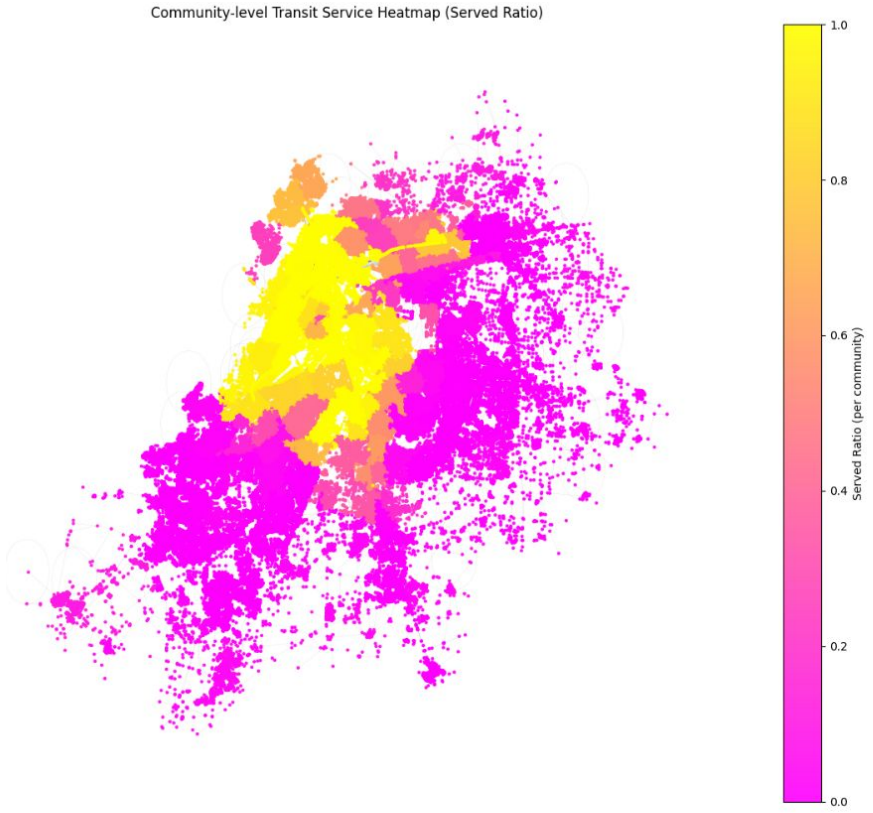
- Label generation (cont.)
  - Benefit of picking obvious nodes: natural train/ test partitioning
  - Which nodes are 'obvious'?
    - Population > Mean population & Distance to stop < 1500; OR
    - Population < Mean population & Distance to stop > 4500
  - The obvious nodes are used for training
  - Every single other node is 'uncertain' and becomes part of the 'test' set of our GNN

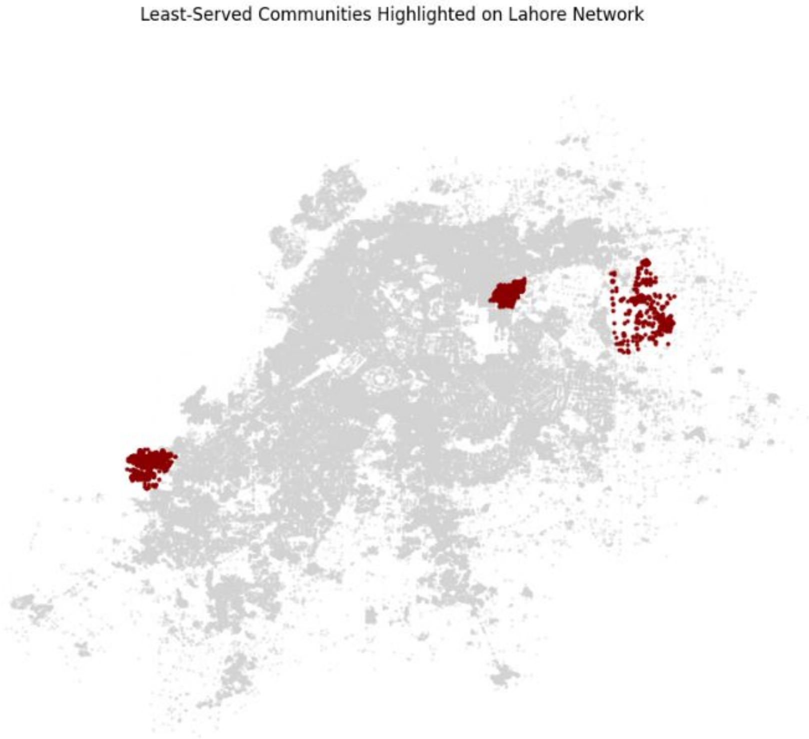

Served vs Unserved Nodes in Lahore (Based on Transit Stops & Population Density)
● Served
● Uncertain
● Low Density Unserved
(C) OpenStreetMap contributors (C) CARTO

Steps (05)

- Final node features: x, y, population, distance_to_nearest_stop, street_count, is_transit_stop, comm_1, comm_2 … comm_k
- Train a simple GCN, and run it on the test set, to get label predictions
- 'Back-project' node classifications onto communities
  - Take a ratio of the number of nodes in that community w/ positively-predicted labels, against the total number of nodes within that community
- Sort all communities using this aforementioned ratio

Results


Community-level Transit Service Heatmap (Served Ratio)
Served Ratio (per community)
1.0
0.8
0.6
0.4
0.2
0.0


Least-Served Communities Highlighted on Lahore Network

Conclusion

- Our results are mildly exaggerated, by virtue of incomplete data, but I do believe that they are a step in the right direction.
- Additionally, I think that more PoIs lying towards the urban center of the city probably skewed the GNN.
- Communities towards the edges of the city are more ‘scattered’ in terms of PoIs, which is not necessarily representative of the truth.

Alternate Approaches (01)

- Community Detection
  - **Label-propagation algorithm** – it worked, but it generated way too many communities.
  - **DBSCAN** – communities were too ambiguous; probably an error on our side, but we decided to not waste time. This would've helped to to *suggest* a route for transit deserts
- Node Classification
  - **GAT** – scrapped due to computational-intensity (1.5hr+ training times)
    - Perhaps because of the abundance of one-hot encoded community labels?
      - May reconsider when we have manual (and thereby fewer + more realistic) community partitions
    - Might have performed better because it learns edge weights & considers neighbouring nodes differently (which is a practical feature in a city network)

Alternate Approaches (02)

An alternative to node classification I concerned was treating this problem as a graph classification problem.

1. Convert Louvain’s generated communities to separate subgraphs
2. Take a few ‘obvious picks’ (manually) and label them accordingly
3. Train a GNN
4. Run it on the remaining/uncertain communities

Logistic Concerns w.r.t Development

- Jupyter notebooks are intrinsically deeply flawed
  - Large heap-allocations during data-wrangling subsequently result in frequent OOMs on older hardware; garbage collection is a sin
  - Irreproducible outputs due to mismanagement of variable state
- Python tooling conflict; `pyproject.toml` files weren't updated
- Version control clobbering w.r.t Jupyter notebook metadata
- Coordinate Reference System (CRS) ambiguity wasted crucial hours and mental fortitude – ESPG:4326 <-> ESPG:32643

Postmortem

- In hindsight, our primary problem was that data preprocessing took way too long.
- We also should have been using a Git hook to wipe out notebook metadata, to prevent needless changes.
- In praise of our project, one extremely powerful observation w.r.t our work is how *easy* it is to scale to other cities.
  - Most ‘modules’ (be it data preprocessing, community detection, or the GNN) are interchangeable (w/ a little hassle to ensure the inputs/outputs line up).

Future Work

Below lie my assumptions/caveats/limitations. These can be ameliorated/removed as future directions of work.

- Few stops; we only have the Red Metro, Orange Line, and 70 unique Speedo stops
- 'Communities' returned by Louvain are not representative of actual residential 'societies' in Lahore
- Don't account for 'self-sufficient' methods of transport e.g communities w/ a teeming qingqi culture
- Missing data e.g stops along Speedo routes (we ignore these)