Graph Neural Network Approach to Census Prediction: Mozambique Case Study

Beate Desmitniece

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2024

Abstract

Up-to-date census information is essential for effective resource allocation, service planning and disaster mitigation. While previous studies have utilised probabilistic modelling, traditional machine learning methods and convolutional neural networks for census estimation, our work introduces a novel approach, employing graph neural networks (GNN) for population count prediction of Mozambique's administrative units. After testing 3 GNN architectures, we found GCN and GAT to be unsuitable for the task, with the GraphSAGE model achieving the best performance, reporting R^2 scores of 85.30% and 87.56% on 2nd and 3rd level administrative unit population predictions, respectively. Notably, GNNs trained on transportation-based edge graphs outperformed those based on geographical adjacency. However, the small dataset size hindered GNNs from effectively learning underlying data patterns, leading them to be outperformed by linear regression and random forest baselines. We used a recently introduced homophily measure *HReg* to investigate whether trends in GNN classification tasks translate to regression tasks, revealing inconsistencies where graphs with high reported homophily underperformed on GCN and GAT architectures, unlike in classification tasks. The models were trained using a self-curated geospatial feature dataset, with nighttime light, building footprints and OpenStreetMap data identified as the most influential predictors.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Beate Desmitniece*)

Acknowledgements

I would like to thank the project supervisor Dr. Sohan Seth for providing guidance, inspiration and support throughout the duration of the project. Working with him has been a great learning opportunity, which has enriched my academic growth. Additionally, I want to acknowledge the help and technical guidance of members of the SHaPE group - Karthik Mohan and Sean Ó Héir.

I would like to give a special thanks to my coursemates Patryk Kuchta and Kieran Swenson for their valuable advice, feedback and emotional support through the development of the project. Lastly, I want to thank my friends Kristers Saulītis and Oliwia Kownacka, who have motivated and encouraged me to strive for excellence in my work.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Access to accurate national census data, specifically population counts, serves a vital role in the decision-making process of governmental bodies. It is used for regional fund allocation, public and private service planning, statistical analysis, social and economic research, as well as vulnerable population identification and disaster effect mitigation (O'Hare, 2019; Jones et al., 2021; Claire, 2007). However, access to such up-to-date information is often restricted, as most of the countries conduct census surveys only every 5 or 10 years due to their high costs and labour-intensive nature, hindering effective resource distribution and policy-making processes (Pelletier, 2020). The challenges associated with the population count acquisition, highlight the need for a more automated and cost-effective solution.

Previous studies have identified geospatial data as an informative source of predictor variables for the task of census prediction Georganos et al. (2022). To perform population count or density estimation, researchers have adopted various methodologies, including probabilistic modelling (Weber et al., 2018), regression models (Engstrom et al., 2020), traditional machine learning approaches (Ahmed et al., 2019) and convolutional neural networks (Neal et al., 2022). However, such methods typically base their population estimates on the properties of the examined area, ignoring the attributes of surrounding regions.

Findings in spatial demography suggest that the population dynamics in the examined region are influenced not only by the region's specific properties but also by the geospatial attributes of the surrounding areas. Several studies have illustrated how various factors such as the increase in manufacturing and service amenities (Diego

Firmino Costa da Silva and da Mota Silveira Neto, 2017), growth in the number of well-being facilities and housing density (Peng et al., 2021), and changes in land development (Tong and Qiu, 2020) cause positive or negative population density changes in the surrounding areas of the examined region. We hypothesise that incorporating the surrounding region attributes, alongside the examined area features, into the predictive census models could lead to more precise population count estimates.

1.2 Proposed Graph Neural Network Population Prediction Pipeline

Building on this hypothesis, our study proposes a population prediction pipeline applicable to any administrative area with finer-grained subdivisions. The pipeline consists of three core modules: graph construction, feature extraction, and predictive modelling (Figure 1.1). In the first module, a graph is constructed using the administrative divisions of the examined region, where vertices represent administrative units, and edges are formed between adjacent units or units interconnected by high-importance transportation networks. In the second module, for each unit/vertex, a set of geospatial features is obtained by processing data from publicly available sources. The use of publicly available data ensures the low cost and ease of reproducibility of the proposed method. In the final module, a graph neural network (GNN) is trained using the constructed graph, and geospatial features of all vertices, but only using the population counts of vertices, where such data is available. Further, predictions are made for the vertices where the population counts are unknown. The GNN model was selected due to its inherent prediction-making that relies on data exchange with its neighbouring regions(Zhou et al., 2022), which aligns with the aforementioned findings that population counts of an area are influenced by the features of its surrounding areas. By iteratively aggregating the feature representations of the neighbouring vertices, we expect the GNN to capture the relationship between the neighbouring areas and make more precise population count estimates.

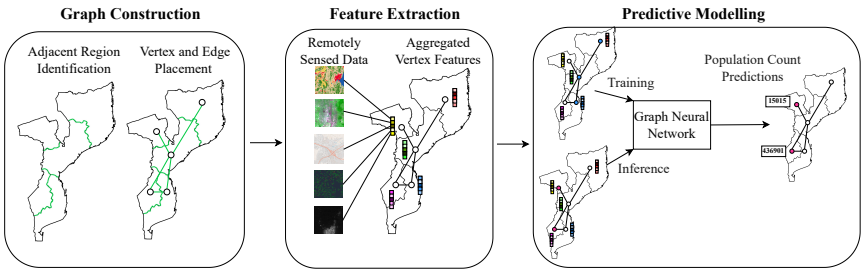

Figure 1.1: Proposed population prediction pipeline, consisting of 3 modules: graph construction, feature extraction and predictive modelling. For the graph construction module, an example of geographical boundary graph construction is provided.

1.3 Application and Evaluation of the Population Prediction Pipeline

To test the proposed pipeline, we applied it to Mozambique at two administrative levels. First, we constructed a graph at the second administrative level, where each vertex represents one of the 159 districts of Mozambique. Alternatively, we created a more fine-grained graph at the third administrative level, where each vertex represents one of the 411 postos (localities) of Mozambique. We characterised each vertex using a variety of geospatial features, including remotely sensed imagery band values, land use data, OpenStreetMap data, building footprint data and nighttime light data (see Chapter 3 for details). It must be emphasised that the dataset used for model training was self-curated.

Our research investigated the use of the 3 most widely adopted GNN architectures-graph convolutional network (GCN) (Kipf and Welling, 2016), graph attention network (GAT) (Veličković et al., 2018) and GraphSAGE (Hamilton et al., 2017) - for the task of population prediction. Previous studies in classification, such as Zhu et al. (2020), have highlighted that the GCN and GAT model performance is highly dependent on the rate of graph homophily - a phenomenon, where edges primarily form between similar vertices. Nonetheless, research on how these architectures perform in regression tasks under varying homophily rates remains sparse. To address this gap, we utilised a recently proposed homophily metric *HReg* (Mueller et al., 2024) to measure the graph homophily in regression tasks and investigated whether the performance trends in classification tasks hold for vertex regression.

Additionally, we are interested in investigating the generalisation ability of the GNN to areas which are geographically distant from those where the trained data was

acquired by developing training/testing splits based on the provinces of Mozambique (first administrative level). Further, since the prediction-making process of the GNN, like most deep learning models, is obscure, we improve its interpretability by analysing which predictor variables play the most significant role in the model's predictions. Lastly, to assess the reliability of the model's predictions and quantify its uncertainty, we constructed confidence intervals, gaining a better understanding of the potential variability in the population count estimates.

1.4 Aim and Objectives

The aim of the study was to develop a population prediction pipeline using graph neural networks by employing geospatial attributes as predictor variables. To achieve this aim, we introduced the following objectives:

1. Construct graphs representing the second and third administrative levels of Mozambique utilising edge types based on both the adjacency of areas and transportation networks (Graph Construction Module).
2. Curate a dataset of publicly accessible geospatial features of Mozambique's administrative units (Feature Extraction Module).
3. Train the GCN, GAT and GraphSAGE models using the developed graphs and geospatial data (Predictive Modelling Module).
4. Assess the performance of the proposed population prediction pipeline through the following sub-objectives:
    - 4.1 Compare the models' performance against baseline models from the previous research literature.
    - 4.2 Evaluate the models' ability to generalise to geographically distant areas by training/testing on province splits.
    - 4.3 Determine the geospatial feature importance by employing permutation feature importance.
    - 4.4 Quantify models' uncertainty by calculating confidence intervals.
5. Assess whether the performance of GNNs in the vertex regression task exhibits the same dependency on graph homophily, measured by the recently proposed *HReg* metric, as observed in classification tasks.

Chapter 2

Related Work

2.1 Census-Dependent Population Prediction

In previous research, the task of census prediction has been primarily tackled via census-dependent or census-independent methods. Census-dependent methods, also known as top-down approaches, utilise the population count of a large-scale region to estimate the population counts of its sub-units. One such approach is dasymetric mapping, where thematic layers are used to assign the weights to the sub-units, allowing for population count disaggregation of the larger area (Xiao Huang and Ning, 2021). Alternatively, a predictive model can be trained to estimate the population count of each sub-unit with the training objective ensuring that their sum matches the population of the large-scale region (Jing Xia and Peng, 2024).

2.2 Census-Independent Population Prediction

Census-independent methods, frequently referred to as bottom-up approaches, develop a prediction model using known population counts of the regions of interest and apply it to estimate the population counts in areas with unknown populations (Neal et al., 2022). In our study, we opt for the census-independent approach since the development of a census-independent model only requires the population counts from a sample of sub-units, whereas the census-dependent approach necessitates the population of the whole examined area to be known beforehand. This makes the census-independent methods applicable to a greater amount of regions, including those which might have no census data associated with them whatsoever.

In previous studies, census-independent methods have relied on various forms of

geospatial data for population prediction, differing in predictive modelling techniques and specific data types. Several studies used probability distribution models for census prediction. For example, Weber et al. (2018) utilised visual attributes from high-resolution satellite imagery to construct a human settlement binary mask and classify each examined block (7.7×7.7 m) into one of eight residential types. A log-normal distribution was modelled for each type to predict population density and aggregates were produced for 93×93 m cells in northern Nigeria. Ma et al. (2024) employed high-resolution satellite imagery band, spectral index, urban morphological and building data aggregates to determine the local climate zone type for each of the examined grid cells (100×100 m) in cities of China. For each of the climate zones and city size type the researchers constructed a log-normal distribution to model the population density, which was then converted into population counts.

Various forms of regression have been most frequently used for population predictive modelling. Engstrom et al. (2020) utilised LASSO-regularised Poisson regression to predict population density in Sri Lankan villages (avg. 4.75 km^2). The predictor variables were the mean nighttime light, tree coverage, topography data, built-up area and geospatial indicators obtained from high-resolution satellite imagery. In another study, Leasure et al. (2020) modelled the population counts of grid cells (100×100 m) in Nigeria using a Poisson process, with population densities estimated by a log-linear regression, where the intercept was obtained hierarchically using data from various size regions. The covariates used for census prediction were WorldPop population estimates, school density, household sizes and area measures of various types of residential locations. Similarly, Boo et al. (2022) applied the same estimation technique but used a different set of predictive variables, extracted from building footprints, and tested their method on grid cells (100×100m) from the Democratic Republic of Congo.

Hillson et al. (2019) applied a linear regression model, obtained through Bayesian Model Averaging, to model the population density in villages (0.2-2.33 km^2) of Bo, Sierra Leone. The input features for each village were the normalised pixel mean, standard deviation, and variance of the satellite imagery bands. Likewise, Neal et al. (2021) applied a linear regression model to predict population counts in grid cells (100×100 m) of Mozambique. They utilised building area extracted from high-resolution satellite imagery, remotely sensed image band values, settlement layer, land cover classification, nighttime light and distance to road data as their predictor variables. Additionally, Yagoub et al. (2024) constructed a geographically weighted regression model to predict the population in the districts (avg. 420 km^2) of Al Ain city. The model utilised mean

nighttime light data, land surface temperature, vegetation index, and building area, with the latter three extracted from high-resolution satellite imagery. Lastly, Ahmed et al. (2019) estimated the refugee population in camp blocks (90×90 m) in Bangladesh using high-resolution satellite band image values and descriptive features of the camp blocks. The researchers experimented with 11 machine learning algorithms, including linear regression, random forest, fully connected neural network and others.

The final category of population prediction models utilises convolutional neural networks, which primarily use aerial imagery to make population estimates. Robinson et al. (2017) trained a VGG-A neural network on Landsat 7 band grid cell (∼15×15 m) imagery to estimate the population in the USA. Alternatively, Neal et al. (2022) trained a ResNet-50 in a self-supervised manner and fine-tuned it on the task of population prediction of grid cells (100×100 m) in Mozambique. The encoded grid cells representation was then parsed through a random forest model to obtain the final population prediction. A similar architecture, ResNet-18, was adopted by Georganos et al. (2022), who used very high-resolution satellite imagery with building footprint imagery to obtain population estimates of grid cells (100×100 m) in Sub-Saharan Africa. Finally, Doda et al. (2024) modified the ResNet-50 architecture to perform population prediction using very high-resolution satellite, nighttime light, land use, local climate zone, elevation images and OSM tabular features to perform population estimation in Europe.

2.3 Building and Expanding on Previous Research

Our study takes inspiration from the previous research on census-independent population estimation methods by utilising data sources and predictor variables in the training of our model as mentioned in several studies, such as Engstrom et al. (2020) and Hillson et al. (2019). For the details of the specific data used in our model training and which studies have shown them to be useful predictor variables, we refer the reader to Chapter 3. Additionally, we employed the linear regression and random forest predictive models as our baselines, since the findings of (Hillson et al., 2019) and Ahmed et al. (2019) have characterised them as suitable for population prediction.

We expand the current research field of models used for population estimation, as no work has previously employed graph neural networks. Lastly, we are one of the few to perform population prediction of large regions (2 and 3-level administrative divisions), as the majority of the existing research has performed estimation on local regions, which are typically only a few m^2 large.

Chapter 3

Data

The following chapter describes the self-composed dataset used for graph construction and model training. We detail the data sources and the processing steps taken to develop the features of the administrative units of Mozambique used in the proposed pipeline's feature extraction module. Table 3.1 illustrates all the predictor and target features used in model training, alongside their source, product, and dates of creation.

3.1 Administrative Boundary Data

Underlying the data aggregation, graph construction and training/testing set creation, was the need for Mozambique's administrative division boundaries. Hence, shapefiles for the 1st (province), 2nd (district), and 3rd (postos) administrative level units were obtained from OCHA Regional Office for Southern and Eastern Africa (ROSEA) (2019). See Figure 3.1 for a visualisation of district and posto administrative boundaries.

3.2 Population Counts

The target variable population counts of districts for August 2023 were obtained from UNFPA (2023). The researchers utilised the country-level census data from 2017 to perform population projections to 2023 and disaggregated it to district-level estimates.

To acquire the target population counts of postos, we used WorldPop Open Population Repository gridded estimates for 2022 (Gadiaga et al., 2023). The researchers produced the estimates by projecting the 2017 district-level census data to 2022 and applied dasymetric mapping to estimate grid cell (100×100 m) population. To obtain the population of a posto, we aggregated all grid cells that fell within the boundary of

the unit. For partially intersected cells, the intersection percentage was calculated and multiplied by the grid cell population to obtain the citizen count of the region.

Although our study uses population estimates instead of true census values as the target variable, the population counts are close approximations of actual citizen counts. The study's main purpose is to test the applicability of GNNs for population prediction, rather than determine the true population of Mozambique's administrative units.

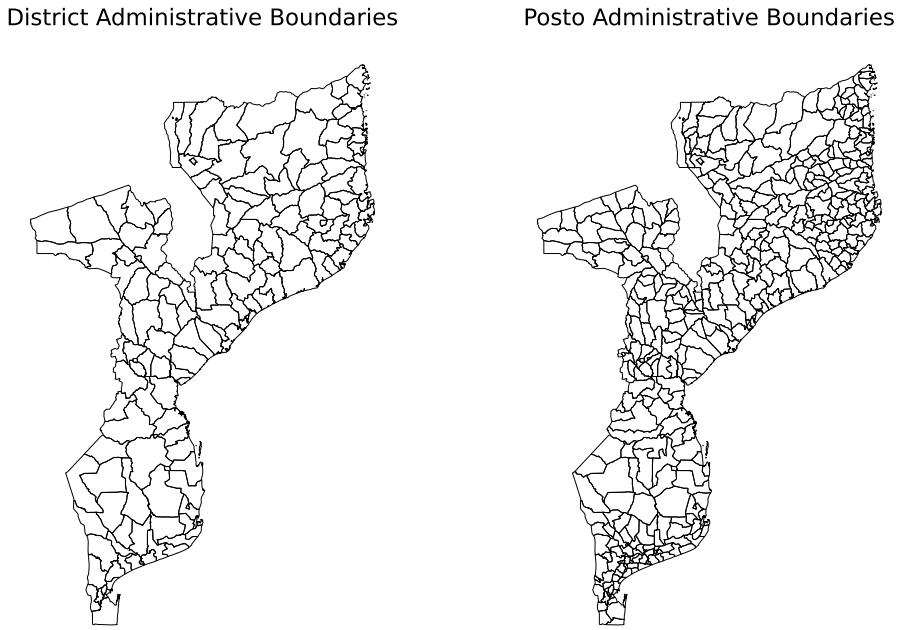

Figure 3.1: Administrative boundaries of Mozambique's 2nd level divisions (districts) and 3rd level divisions (postos).

Features	Feature Count	Source	Product	Date
Population Counts	Target	Gadiaga et al. (2023)	WorldPop	2022
		UNFPA (2023)	Mozambique Data Grid	2023
District area (km^2)	1	UNFPA (2023)	Mozambique Data Grid	2023
Land Cover Distribution of water, trees, flooded vegetation, crops, built area, bare ground, snow/ice, rangeland	8	Karra et al. (2021)	Sentinel-2 10m Land Use/Land Cover Time Series	2022, 2023
Mean, standard deviation, variance of surface reflectance bands (1-7), temperature, radiance (thermal, upwelled, downwelled), atmospheric transmittance, estimated emissivity, NDVI, NDWI, NDBI	48	U.S. Geological Survey	Landsat 8-9 Collection 2 Level 2 Science Product	2019-2024
Counts of POIs, places of worship, localities, traffic-related POIs, transport-related POIs. Length of railways, roads.	7	OpenStreetMap contributors (2017)	OpenStreetMaps	1/1/2023, 1/1/2024
Building count and area	2	Sirko et al. (2021)	Open Buildings	Unknown
		Microsoft (2024)	Bing Maps	2012-2024
Mean, standard deviation, variance of nighttime lights	3	Román et al. (2018)	VNP46A1 - VIIRS/NPP Daily Gridded Day Night Band 500m	October 2022, June 2023

Table 3.1: Predictor and target variables used for model training, alongside their sources, products, and creation dates. All features were normalised before training.

3.3 Geospatial Features

3.3.1 Land Use Features

As shown by Engstrom et al. (2020), land cover indicators, like the extent of built-up areas, are important determinants for population prediction in low-population regions. Hence, we obtained the land use imagery for years 2022 and 2023, provided by Karra et al. (2021). The dataset contained a 10-m spatial resolution map with each pixel assigned a land use class: water, trees, flooded vegetation, crops, built area, bare ground, snow/ice, clouds, and rangeland. The map was generated using predictions from a UNet convolutional neural network, which was trained on the RGB pixel classification task.

For each posto and district, we calculated the percentage coverage of each class, using data from 2022 and 2023 respectively. The 'cloud' class pixels were excluded from calculations, removing 0.0025% of 2022 data and 0.0019 % of 2023 data.

3.3.2 Landsat 8-9 Features

Inspired by Hillson et al. (2019), who used Landsat 5 images for population estimation, we acquired a similar set of predictive variables from a more recent release - the Landsat 8-9 (courtesy of the U.S. Geological Survey), containing 30-m spatial resolution map of 20 bands. The data originates from 2019 to 2024, with most tiles from 2023. We prioritised tiles with 0% cloud cover over the ones that matched the population count year. However, this constraint was not fulfilled, as some tiles contain up to 5% cloud cover. Pixels with high-confidence clouds, aerosols or surface temperature uncertainty above 5° were masked, resulting in 10.33% of the country being masked.

For each district and posto, we computed the mean, standard deviation and variance of the 7 surface reflectance bands (ultra blue, blue, green, red, near-infrared, shortwave infrared 1 and 2). Additionally, we performed the same calculation using the 6 surface temperature bands- temperature, thermal radiance, upwelled radiance, downwelled radiance, atmospheric transmittance, and emissivity.

Following the approach of Ma et al. (2024) and Yagoub et al. (2024), we used Landsat band imagery to calculate the Normalized Difference Vegetation Index (NDVI), Water Index (NDWI), and Built-up Index (NDBI). The index calculations are as follows:

$$\text{NDVI} = \frac{B_5 - B_4}{B_5 + B_4}, \quad \text{NDWI} = \frac{B_3 - B_5}{B_3 + B_5}, \quad \text{NDBI} = \frac{B_6 - B_5}{B_6 + B_5}$$

B_{3-6} are the green, red, near-infrared and the first shortwave infrared bands. For each district/posto, we computed the mean, standard deviation, variance of the indices.

3.3.3 OpenStretMap Features

As seen in the studies of Leasure et al. (2020), Neal et al. (2021), Doda et al. (2024), features extracted from OpenStreetMaps (OSM) are often used as predictor variables for population estimation. OSM is a publicly accessible geographical database developed by volunteer contributors, containing various geographical features of maps (OpenStreetMap contributors, 2017). We utilised extracts from Mozambique's map to calculate the length of roads and railways, number of POIs, places of worship, localities, traffic-related POIs and transport-related POIs for each district and posto. Although OSM maps provide more detailed classifications, their limited presence in the maps of Mozambique forced us to rely on general classes. We used map extracts from 1/1/2023 for postos, and 1/1/2024 for districts, reflecting the map's state in the preceding year.

3.3.4 Building Footprint Features

Leasure et al. (2020), Neal et al. (2021), and Boo et al. (2022) have all identified building area as a valuable covariate for census modelling. Motivated by their approach, we obtained the building footprints provided by Microsoft, which were generated using a deep neural network trained on the task of semantic segmentation using satellite imagery from 2014-2024 (Microsoft, 2024). Additionally, we sourced building footprints provided by Google (Sirko et al., 2021), which were generated using a UNet model trained on the task of semantic segmentation, excluding those with confidence less than 75%. We are aware that the building footprints do not exactly match the years when population counts were acquired. However, we consider them to be a close enough proxy as changes in urban environments typically occur gradually (Balk et al., 2018).

For all districts and postos, we calculated building counts using both Microsoft and Google footprints. In areas, where one of the sources had unusually low values due to missing data, we selected the maximum between the two. We then calculated the total building area based on the selected count source.

3.3.5 Nighttime Light Features

Following the successful use of nighttime light imagery data of Engstrom et al. (2020), Neal et al. (2021), and Yagoub et al. (2024) in developing census prediction models, we collected 500-m spatial resolution nighttime radiance imagery from the Visible Infrared Imaging Radiometer Suite, VNP46A1 product, develop by (Román et al., 2018). We acquired daily imagery for September- October 2022 and June 2023, as these months had the least cloud coverage for the respective years. For each administrative unit, we retained the image with the least cloud coverage, resulting in 0.1697% of the area having cloud coverage for postos and 0.4193% of the district area being covered by clouds which were excluded from calculations. For each posto/district, we calculated the mean, standard deviation and variance of the nighttime radiance values.

Chapter 4

Methodology

The following chapter begins by discussing the proposed population prediction pipeline's first and third modules: graph construction and predictive modelling. Further, we introduce the baseline models and discuss the details model of training and evaluation. Lastly, we examine the homophily rates of the constructed graphs and their expected performance on GNNs.

4.1 Graph Construction

The first module of the population prediction pipeline and development of any graph neural network begins with graph construction. GNNs utilise graph-structured data for model training and prediction. We denote a graph as $G = (V, E)$, where V represents the vertices in the graph, and E denotes the edges connecting the vertices.

In our study, we experimented with two types of vertices based on administrative division granularity - district vertices and posto vertices. We denote a district vertex as $v_d^D \in V^D$, where V^D is the set of all district vertices in Mozambique, with d representing a specific district. Similarly, a posto vertex is denoted as $v_p^P \in V$, where V_p^P is the set of all postos vertices in Mozambique, with p representing a specific posto. Each vertex i is associated with a feature vector $h_i \in \mathbb{R}^{69}$, containing 69 features.

Additionally, we experimented with two types of edge connections. First, for each type of administrative vertices V^D and V^P, we constructed an undirected edge $e_{ij}^B \in E^B$ between vertices v_i and v_j that shared a geographical boundary. E^B denotes the set of all boundary edges for an administrative granularity level. To develop a connected graph, for regions located in water, manual edges to the 2 nearest regions were added to ensure connectivity.

Second, for each type of administrative vertices V^D and V^P, we constructed an undirected edge $e_{ij}^T \in E^T$ between vertices v_i and v_j that were connected by a major transportation route. E^T denotes the set of all transportation connectivity edges for an administrative granularity level. We focused only on significant transportation routes, including railways, and primary and secondary national roads. The road data was acquired from the OpenStreetMap 2023 extract. The constructed graph contained several isolated vertices, resulting in a disconnected graph.

In total, we had constructed 4 graphs: district vertex and geographical boundary edge graph $G_1 = (V^D, E^B)$ (Figure 4.1), district vertex and transportation connectivity edge graph $G_2 = (V^D, E^T)$ (Figure 4.2), posto vertex and geographical boundary edge graph $G_3 = (V^P, E^B)$ (Figure 4.3), and posto vertex and transportation connectivity edge graph $G_4 = (V^P, E^T)$ (Figure 4.4). All the graphs are undirected, and homogeneous, with their vertices having the same feature set. We refrain from assigning features to the graph edges.

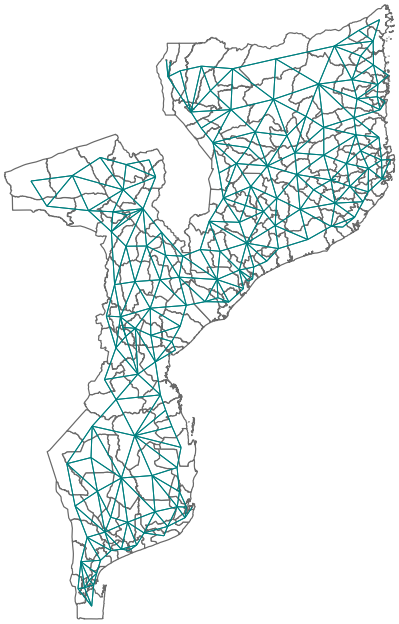

Figure 4.1: Graph of Mozambique districts, connected based on geographical boundary.

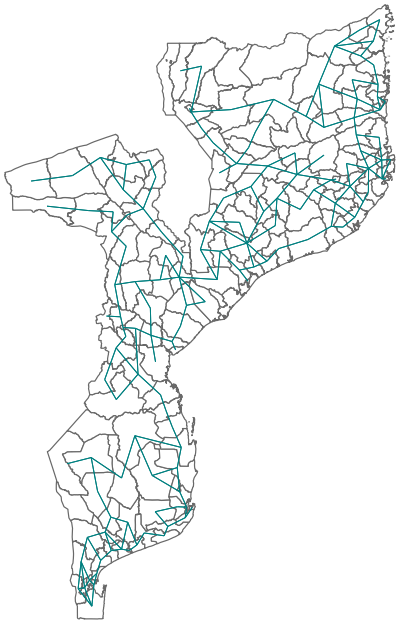

Figure 4.2: Graph of Mozambique districts, connected based on major transportation routes.

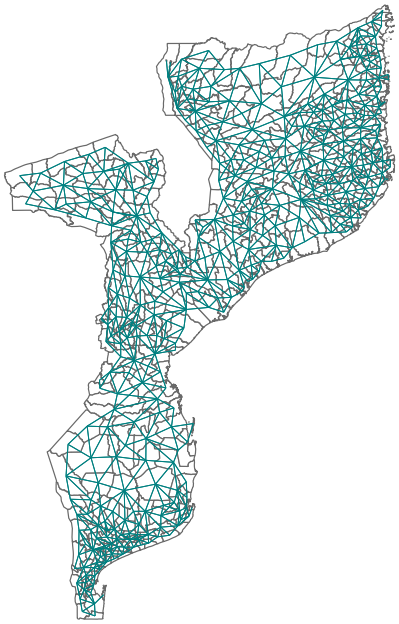

Figure 4.3: Graph of Mozambique postos, connected based on geographical boundary.

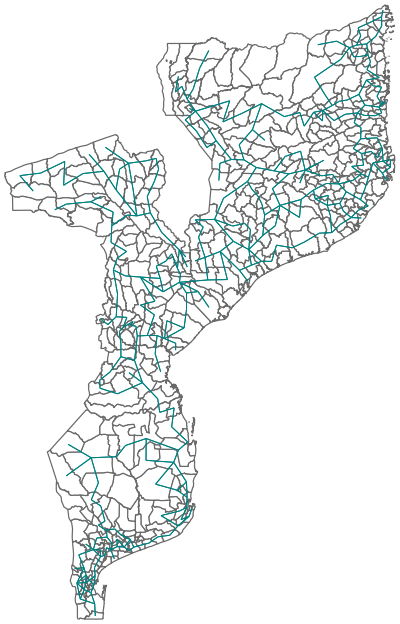

Figure 4.4: Graph of Mozambique postos, connected based on major transportation routes.

4.2 Predictive Modelling - Graph Neural Network

The third module of the population prediction pipeline involves the training and inference of the graph neural network. Due to the vast variation in graph structures and prediction tasks, the design of the training and inference process of graph neural networks is highly adaptable. Therefore, we limit the explanation of GNNs to undirected, homogeneous graphs with vertex-only features applied to the task of semi-supervised vertex-level regression.

In this context, the GNN takes as input the constructed graph G along with its vertex feature H vectors. The graph is parsed through the message-passing layer, where new vertex representations are produced by aggregating the vertex's feature vector with those of its neighbouring vertices. By stacking multiple message-passing layers, the GNN enables the vertex representation to incorporate information from vertices that are not directly connected by an edge. The method of vertex aggregation depends on the specific GNN architecture employed. After the hidden representations of the vertices are obtained, they are passed through a regressor, a fully connected layer in our case, to make a continuous value prediction.

In the semi-supervised setting, the model has access to the features of all vertices

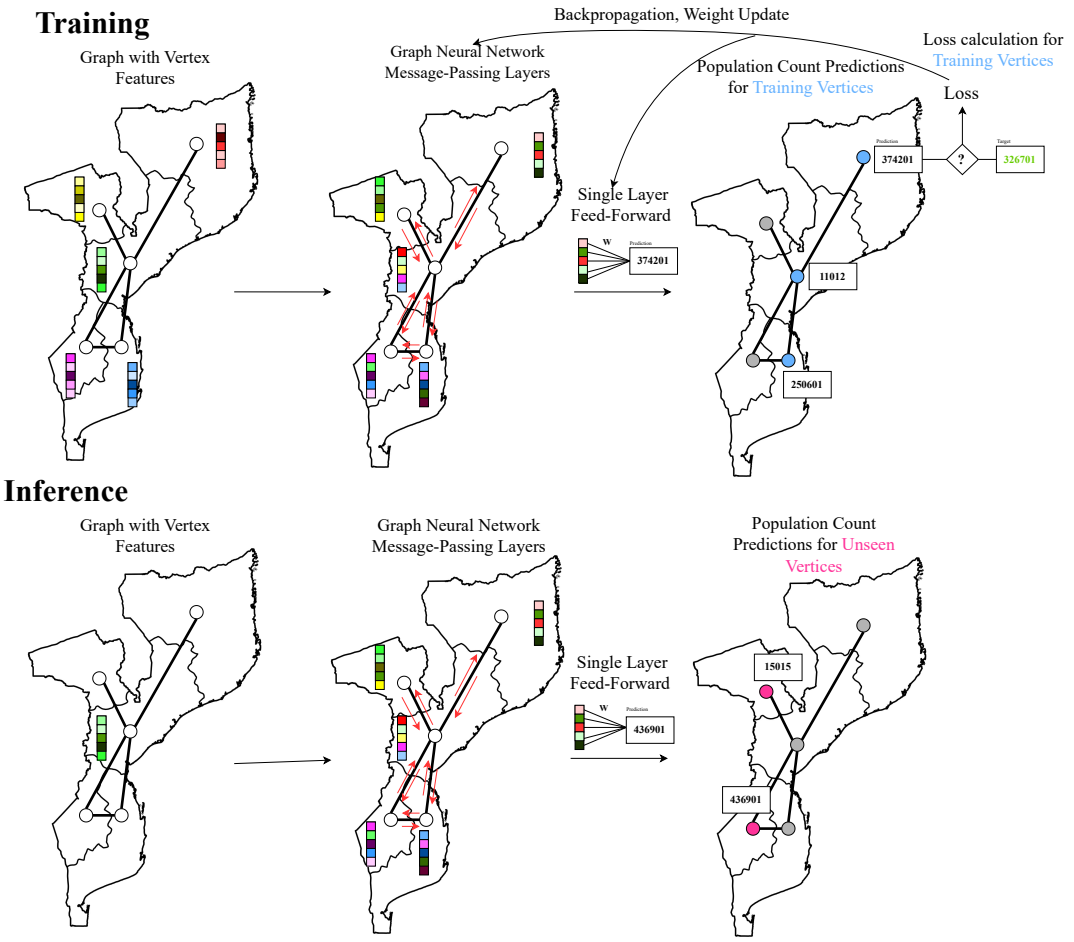

Figure 4.5: Training and inference of the proposed GNN architecture.

in the graph, but only to a subset of their labels. Hence, the loss calculation and backpropagation are performed using only the training set vertices labels, whereas inference and model assessment are performed using the testing set vertices labels.

Figure 4.5 illustrates the training and inference processes of the proposed graph neural network architecture.

4.2.1 Graph Convolutional Networks

The first GNN architecture that we introduce for the task of population prediction is the graph convolutional network (Kipf and Welling, 2016). The GCN model follows the general GNN architecture, defining a distinctive message-passing layer. Within the layer, the vertex representation vector is summed with those of its adjacent vertices. Subsequently, the aggregated vector is normalised, followed by a linear and a non-linear ReLU transformation, resulting in an updated vertex representation. Below we provide

a detailed mathematical description of the GCN message-passing layer.

Initially, an adjacency matrix $A \in \mathbb{R}^{n \times n}$ is constructed based on the input graph, where n is the number of vertices. To ensure that the feature vector of the vertex itself is included during message-passing aggregation, we introduce self-loops to the graph as such: $\tilde{A} = A + I$, where I is the identity matrix and $\tilde{A} \in \mathbb{R}^{n \times n}$.

Further, we construct a degree matrix $D \in \mathbb{R}^{n \times n}$ for the self-loop adjacency matrix $\tilde{A}$, where D_{ij} is equal to the number of neighbouring vertices if $i = j$ and 0 otherwise. We then calculate $D^{-0.5}$ by taking the reciprocal of the square root of each non-zero entry in D. This matrix is used to normalise $\tilde{A}$ as follows:

$$\bar{A} = D^{-0.5}\tilde{A}D^{-0.5} \tag{4.1}$$

The normalised self-loop adjacency matrix $\bar{A}$ is multiplied by the features of the graph vertices at layer l, denoted by $H^{(l)} \in \mathbb{R}^{n \times d}$, where d is the number of features per vertex. The operation calculates the weighted sum of the features of the vertex and its neighbours, with the weights originating from $\bar{A}$:

$$\bar{H^{(l)}} = \bar{A}H^{(l)} \tag{4.2}$$

Lastly, the aggregated features are multiplied by a trainable weight matrix $W^{(l)} \in \mathbb{R}^{d \times w}$, where w represents the dimension of the new vertex representation. After which a non-linear activation function σ is applied, which, as per the original GCN paper (Kipf and Welling, 2016), is the ReLU function. As a result, we have obtained a matrix with the new representations of the graph vertices:

$$H^{(l+1)} = \sigma(\bar{H^{(l)}}W^{(l)}) \tag{4.3}$$

Alternatively, we can express the complete message-passing layer transformation of a graph convolutional network using a single equation:

$$H^{(l+1)} = \sigma\left(D^{-\frac{1}{2}}(A+I)D^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{4.4}$$

4.2.2 Graph Attention Networks

Additionally, we employ an alternative GNN architecture known as the graph attention network (Veličković et al., 2018). The GAT architecture extends the functionality of the message-passing layer by integrating an attention mechanism that weighs the influence of the neighbouring vertices. For each vertex, the GAT computes the attention coefficients between itself and all the adjacent vertices using a single-layer feed-forward

neural network, followed by a softmax normalisation. The features of the vertices are linearly transformed, weighted by the respective attention coefficients, and summed, obtaining an updated representation of the vertex. The attention mechanism enables variable significance to be assigned to each neighbouring vertex, as opposed to the GCN architecture, which presumes an equal contribution from all its neighbours.

Further, a mathematical explanation of the GAT message-passing layer is presented. First, the feature vectors $h^{(l)} \in \mathbb{R}^d$ at layer l of vertex i and its neighbouring vertices $j \in \mathcal{N}(i)$ are transformed by a shared learnable weight matrix $W^{(l)} \in \mathbb{R}^{w \times d}$. The transformation is computed as follows:

$$\bar{h^{(l)}} = W^{(l)} h^{(l)} \tag{4.5}$$

Next, the obtained parameterised representations of the vertex $\bar{h_i^{(l)}}$ and its neighbour representation $\bar{h_j^{(l)}}$ are concatenated and parsed through a single-layer feed-forward neural network parameterised by $a \in \mathbb{R}^{2w}$, with a LeakyReLU activation function ($\alpha = 0.2$) to compute the raw attention coefficient:

$$e_{ij} = LeakyReLU(a^T [\bar{h_i^{(l)}} || \bar{h_j^{(l)}}]) \tag{4.6}$$

Further, a softmax normalisation is applied to the raw attention score:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{t \in \mathcal{N}(i) \cup i} exp(e_{it})} \tag{4.7}$$

Lastly, the updated vertex representation for vertex i is computed by summing the parameterised vertex vector with its parameterised neighbour vertices, weighted by the respective attention coefficients. Lastly, a non-linear transformation σ (ReLU) is applied:

$$h_i^{(l+1)} = \sigma\Big(\sum_{j \in \mathcal{N}(i) \cup i} \alpha_{ij} W^{(l)} h_j^{(l)} \Big) \tag{4.8}$$

4.2.3 GraphSAGE

Lastly, we employed the GraphSAGE architecture for the task of population prediction, as proposed by Hamilton et al. (2017). The message-passing layer of the GraphSAGE network aggregates the neighbouring vertices' features and concatenates the aggregate with the vertex's original representation vector. After this, linear and non-linear (ReLU) transformations are applied to the concatenated vector, producing an updated vertex representation vector.

In contrast to the GCN and GAT architectures, GraphSAGE does not incorporate the vertex's own representation in the aggregation when generating its new representation. Instead, GraphSAGE concatenates the vertex's representation to the aggregated neighbour feature vector. Such an approach provides greater expressive power, allowing the network to prioritise the vertex's own features, when the neighbouring nodes provide a limited signal for accurate prediction-making.

We provide a mathematical explanation of the GraphSAGE message-passing layer. For each vertex i, the aggregate $n_i^{(l)} \in \mathbb{R}^d$ of its neighbouring vertices $j \in \mathcal{N}(i)$ vector representations $h_j^{(l)} \in \mathbb{R}^d$ at layer l is calculated as follows:

$$n_i^{(l)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in N(i)} h_j^{(l)} \tag{4.9}$$

Next, the aggregated vector $n_i^{(l)}$ is concatenated with the vertex's own vector representation $h_i^{(l)}$, resulting in $c_i^{(l)} \in \mathbb{R}^{2d}$:

$$c_i^{(l)} = h_i^{(l)} || n_i^{(l)} \tag{4.10}$$

Lastly, the concatenated vector $c_i^{(l)}$ is linearly transformed by applying a weight matrix $W^{(l)}\mathbb{R}^{d \times 2d}$ and a non-linear activation function σ, resulting in an updated vertex i representation $h_i^{(l+1)} \in \mathbb{R}^d$:

$$h_i^{(l+1)} = \sigma(W^{(l)} c_i^{(l)}) \tag{4.11}$$

4.3 Baseline Models

To gauge the performance of the graph neural network models on the task of population prediction, we proposed four baseline models for comparison.

Due to the frequent use of the **linear regression** model in previous population prediction studies (Hillson et al., 2019; Leasure et al., 2020; Neal et al., 2021), it was selected as the first baseline model. The vertex features were used as predictor variables and the population count of a vertex was used as a target variable. Since linear regression does not inherently support graph structures, edge information was omitted from the model. Model parameters were fitted using ordinary least squares (OLS).

To address the potential nonlinear relationship between the predictor variables (vertex features) and the target variable (population count), we utilised the **random**

forest regressor as the second baseline model. Similar to the linear regression model, we excluded the edge data from the model.

As the final baseline models, we trained the **GNN architectures, modified to exclude any edge information**, using an empty graph (a graph with no edges).

In the case of the GCN message-passing layer, the use of the empty graph implies $A = 0$, simplifying the transformation to:

$$H^{(l+1)} = \sigma\left(D^{-\frac{1}{2}}(A+I)D^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) = \sigma\left(I(0+I)IH^{(l)}W^{(l)}\right) = \sigma\left(H^{(l)}W^{(l)}\right) \tag{4.12}$$

In the GAT message-passing layer, the only element in the vertex neighbourhood is the vertex itself. Hence, the expression can be rewritten as:

$$h_i^{(l+1)} = \sigma(\sum_{j\in\mathcal{N}(i)\cup\{i\}} \alpha_{ij}W^{(l)}h_j^{(l)}) = \sigma(\sum_{j\in\{i\}} \alpha_{ij}W^{(l)}h_j^{(l)}) = \sigma(W^{(l)}h_i^{(l)}) \tag{4.13}$$

As for the GraphSAGE message-passing layer, the averaging step of the neighbouring vertex vectors is omitted due to their absence. Consequently, the concatenation step is excluded as well, simplifying the vertex representation update to:

$$h_i^{(l+1)} = \sigma(W^{(l)}[h_i^{(l)}||NULL]) = \sigma(W^{(l)}h_i^{(l)}) \tag{4.14}$$

The modified transformations of the graph neural networks resemble a fully connected feed-forward network layer. The only deviation occurs in the modified GCN layer, which assigns a unique set of weights to each vertex, rather than having shared weights for all vertices. Hence, as the baseline models we select the GCN and GAT architectures, trained on empty graphs. The empty graph GraphSAGE model derivation equals the GAT model one, making its selection redundant.

Training an empty graph neural network provides a reference point for performance comparison with models trained on non-empty graphs. The baseline models will allow us to assess the impact of the graph structure on the model performance.

4.4 Model Evaluation

4.4.1 Cross-Validation

Due to the small size of the datasets employed in the study (159 district samples, 411 postos samples), as suggested by Raschka (2018), to obtain a reliable model performance estimate, we employed k-fold cross-validation.

For each fold, a model was trained using data from $k-1$ folds, with predictions obtained from the k^{th} fold. Model evaluation was conducted by pooling all predictions from all folds and calculating the evaluation metrics on the pooled dataset.

To assess the models' ability to generalise across various administrative divisions, we constructed 2 cross-validation setups. In the first setup RS, we randomly divided the data samples into $k = 10$ folds to test how well the model generalises to previously unseen regions of the same administrative level. In the second setup GS, the samples were split into $k = 11$ folds by assigning them to one of the 11 respective Mozambique provinces (1^{st} administrative level). The geographical cross-validation split allows us to assess how well the model generalises to areas that are geographically distinct and belong to a different higher-level administrative unit. Both setups resemble a real-world scenario, where data is available for certain administrative regions of the country, but predictions are required for other areas, where such data is unavailable.

4.4.2 Evaluation Metrics

To gauge the efficacy of the introduced graph neural networks and baseline models, we utilised R^2 score, which indicates the proportion of variance in the target variable (population count) that can be predicted by the model. It is calculated by:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_i (y_i - \bar{Y})^2} \tag{4.15}$$

n represents the number of data samples, y_i is the target value, $\hat{y}_i$ is the model's predicted value and $\bar{y}$ is the mean of the target values.

Additionally, we calculated the mean absolute error (MAE) and median absolute error ($MeAE$), with the latter being more robust to outliers:

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \qquad MeAE = \text{median}(|y_1 - \hat{y_1}|, ..., |y_n - \hat{y_n}|) \tag{4.16}$$

Additionally, the mean absolute percentage error ($MAPE$) and median absolute percentage error ($MeAPE$) were employed, with the latter being more robust to outliers:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \qquad MeAPE = \text{median}\left(\frac{|y_1 - \hat{y_1}|}{y_1}, \ldots, \frac{|y_n - \hat{y_n}|}{y_n}\right) \tag{4.17}$$

The choice of evaluation metrics was heavily inspired by the work of Neal et al. (2022), who utilised the R^2 score, $MEAE$ and $MeAPE$ in the assessment of their population prediction model.

4.4.3 Feature Importance

To assess which of the predictor variables have the greatest contribution to the model's performance, we applied the permutation feature importance algorithm, as proposed by Breiman (2001).

The method calculates the original prediction error on the pooled cross-validation set E_{orig} (MAE was used in our analysis). After, the values of feature i in the test set are shuffled, obtaining new predictions for each cross-validation fold. The newly acquired predictions are pooled together and error E_{perm} is recomputed. Then, the feature importance quotient is calculated as $FI_i = \frac{E_{perm}}{E_{orig}}$.

A high FI_i value suggests that the permutation of feature i negatively impacts the model performance, making it of great importance. Conversely, a low FI_i value suggests minimal impact, indicating that the feature is not critical for prediction-making.

4.4.4 Confidence Intervals

To provide a greater understanding of how certain the models are in their predictions, we quantised their uncertainty using confidence intervals. The confidence interval illustrates the range of values within which the model believes the true value lies, with a certain level of confidence (95% in our case). They can act as a measure of the model's reliability, where models with small confidence intervals are considered more certain in their predictions, while large confidence intervals indicate greater uncertainty. The confidence intervals were computed using the pooled cross-validation predictions.

To obtain confidence intervals for GNN model predictions, we first employed Monte Carlo dropout (Gal and Ghahramani, 2016). Monte Carlo dropout enables dropout during inference and acquires predictions, interpreted as predictive distribution samples. This method views dropout as an approximation of Bayesian inference in deep Gaussian processes, where the posterior distribution over model weights is modelled by enabling dropout during inference. Since the Monte Carlo dropout authors conducted experiments on a relatively small neural network and dataset of similar size to ours, we consider it an appropriate method for confidence interval calculation.

After having obtained the samples from the model's predictive distribution, we apply the parametric confidence interval estimation for the mean and calculate the confidence interval CI for i sample as:

$$\begin{aligned} SE &= \frac{\sigma}{\sqrt{n}} \\ CI_i &= \bar{y}_i \pm z \times SE \end{aligned} \tag{4.18}$$

σ is the standard deviation of all the predictions, n is the number of Monte Carlo dropout runs (100 in our case), $\bar{y}$ represents the mean of the predictions, z is the critical value of the confidence level (1.96 in our case).

In the case of the random forest model, we treat the prediction of each tree as a predictive distribution sample and calculate the *CI* as shown in Equation 4.18.

For the OLS linear regression baseline, we utilised the Gauss Markov theorem best linear unbiased estimator (Seber and Lee, 2003) and calculated the confidence interval for i sample as:

$$\begin{aligned} SE_i &= \sqrt{\sigma^2 x_i^T (X^T X)^{-1} x_i} \\ CI_i &= \hat{y}_i \pm z \times SE_i \end{aligned} \tag{4.19}$$

σ^2 represents the training set residual variance, $X \in \mathbb{R}^{n\times(d+1)}$ is a matrix containing feature vectors of the training set with n samples and d features, including a column of ones for the intercept, $x_i \in \mathbb{R}^{d+1}$ is the feature vector of an unseen sample and $\hat{y}_i$ is the predicted value of an unseen i sample.

4.5 Homophily and Graph Neural Network Performance

Graph neural network performance is often studied in relation to the level of homophily present in the graph data, which refers to a tendency of edges to form between vertices that have similar features or share the same label. Homophily is typically measured in a $h \in [0,1]$ range, where 0 indicates low homophily (edges formed between dissimilar nodes) and 1 indicates high homophily (edges formed between nodes with the same label). Previous research on classification tasks has noted that for certain GNN architectures, such as GCN and GAT, to achieve viable results, they must be trained on graphs with high homophily rates ($h = 0.7$)(Zhu et al., 2020). This is because the vertex representations are calculated by taking an average or a weighted average of the neighbouring vertex vectors, including the vertex's own feature vector. As a result, the neighbouring vertices tend to have similar representations, implying that their labels must also be alike. On the other hand, architectures like GraphSAGE are less affected by low-homophily graphs ($h = 0.1$) as they exclude the vertex's own feature representation from aggregation, making them robust to dissimilar neighbours.

While extensive research has focused on the influence of homophily on GNN performance in classification tasks (Zhu et al., 2020; Platonov et al., 2024), the evaluation of homophily in vertex regression tasks has remained largely unexplored. To assess the

homophily rate for graphs with continuous vertex labels, we utilise the *HReg* metric introduced in a recent study by Mueller et al. (2024). The formula of the metric calculation is provided below, where $G = (V,E)$ represents the graph as defined earlier in Section 4.1, $\mathcal{N}^k(i)$ represents the neighbours of vertex i reachable within k hops (message-passing layers), and Y is the vector containing vertex labels:

$$HReg(G,Y) = 1 - \left(\frac{1}{|V|} \sum_{i \in V} \left(\frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}^k(i)} |y_i - y_j| \right) \right) \tag{4.20}$$

The *HReg* homophily measure of the 4 introduced graphs is presented in Table 4.1 with k values up to 8. We limit k to 8 as previous studies have shown that the increase in message-passing layers results in over-smoothing, where the vertex representations become too similar, losing the predictive signal (Rusch et al., 2023). All graphs indicate a high homophily rate, with values exceeding 0.86 and homophily rates decreasing as k increases. The posto-level graphs exhibit *HReg* values that are more than 0.04 higher than the district-level graph ones, likely due to the finer granularity of administrative levels having less population count variation between neighbouring regions.

Our study aims to investigate whether the recently proposed *HReg* homophily metric values align with the performance trends observed in classification tasks. More specifically, as suggested by the high *HReg* homophily indicator, we expect the GCN and GAT model performance to match that of GraphSAGE.

k-hops	$G_1 = (V^D, E^B)$	$G_2 = (V^D, E^T)$	$G_3 = (V^P, E^B)$	$G_4 = (V^P, E^T)$
1	0.9172	0.9283	0.9513	0.9604
2	0.8952	0.9094	0.9415	0.9511
3	0.8861	0.9002	0.9382	0.9471
4	0.8808	0.8947	0.9351	0.9443
5	0.8763	0.8909	0.9344	0.9434
6	0.8721	0.8899	0.9337	0.9432
7	0.8703	0.8884	0.9328	0.9426
8	0.8696	0.8871	0.9315	0.9419

Table 4.1: *HReg* graph homophily values of the G_1 (district vertices, geographical boundary edge graph), G_2 (district vertices, transportation connectivity edge graph), G_3 (posto vertices, geographical boundary edge graph), G_4 (posto vertices, transportation connectivity edge graph) for various k-hop values.

Chapter 5

Results and Discussion

5.1 Hyperparameter Search

To find the most optimal hyperparameter set, an exhaustive grid search was conducted when training the GCN, GAT and GraphSAGE models. We experimented with the following hyperparameters: hidden layer size $\in \{2,4,8,16,32,64,128,256\}$, message exchange count $\in \{1,2,3,4,5,6,7,8\}$, learning rate $\in \{0.001,0.01,0.1\}$, dropout rate $\in \{0,0.05,0.1,0.15,0.2,0.25,0.3\}$, and epochs up to 600. For the random forest training, we experimented with the number of minimum samples required to split a node $\in \{2,2,4,5,6,7,8,9,10\}$. For each hyperparameter combination, k cross-validation models were trained, and the cross-validation *MAE* was computed using the pooled predictions from the validation folds. The hyperparameter set yielding the lowest *MAE* was selected as the final one. The final hyperparameter sets used for all model training and evaluation are available in Appendix A.2.

5.2 Model Performance Results

For each baseline and GNN architecture, we trained 4 models to assess the models' prediction-making and generalisation performance across various administrative levels. 2 models were developed for population prediction on a district level (2^{nd} administrative level): one used randomly assigned cross-validation splits, whereas the other utilised splits based on the geographical grouping of districts within a province. Similarly, the other 2 models were trained to perform population prediction on a posto level (3^{rd} administrative level): one using randomly assigned cross-validation splits, whereas the other employed splits based on the geographical grouping of postos within a province.

5.2.1 Baseline Results

		District-level					Posto-level				
		MAE	*MeAE*	*MAPE*, %	*MeAPE*, %	R^2, %	*MAE*	*MeAE*	*MAPE*, %	*MeAPE*, %	R^2, %
Linear Regression	RS	50939.59	32607.28	30.58	22.47	87.59	28946.93	15948.37	311.27	39.79	68.47
	GS	72395.94	47502.15	39.94	27.83	62.10	32021.93	16325.89	1610.52	41.73	74.48
Random Forest	RS	57926.02	36977.20	31.35	23.87	77.81	28176.00	13920.05	199.57	36.03	77.17
	GS	63390.59	40664.09	37.48	23.44	77.77	29391.42	14347.76	287.46	37.09	78.66
GCN, empty graph	RS	62700.67	43765.49	35.79	23.28	78.70	29766.48	17033.84	341.53	45.28	84.10
	GS	77404.50	56160.16	56.25	31.83	68.53	33235.95	18252.14	732.35	48.21	80.95
GAT/GraphSAGE, empty graph	RS	63113.81	43433.63	41.52	29.60	81.52	29196.65	17696.21	199.37	49.66	79.91
	GS	74227.35	48858.94	52.07	32.86	74.67	31172.38	20689.95	1036.71	56.11	70.19

Table 5.1: Cross-validation evaluation metrics for baseline models: linear regression, random forest, GCN (empty graph), and GAT/GraphSAGE (empty graph). For each baseline type district-level and posto-level prediction making models were trained. Each model's performance was assessed using two different cross-validation methods: random splits (RS) and geographical splits (GS).

The evaluation metrics of all baseline models are illustrated in Table 5.1.

Out of the linear regression models trained for district population prediction, the random split model significantly outperformed the geographical split one, achieving an R^2 score of 87.59%, compared to the geographical split's R^2 score of 62.10%. A similar occurrence was observed for posto-level prediction models, where the random split model achieved a *MAE* of 28946.94 but a slightly lower R^2 score of 68.47% compared to the geographical split's *MAE* of 32021.93 and R^2 score of 74.48%. The observed phenomenon, where models yield better results when trained with random splits than geographical splits, can be attributed to the significant differences between provinces. The models were unable to generalise effectively to the unseen areas since the training and testing data were drawn from different distributions. Additionally, lower *APE* metrics at the district-level, such as *MAPE* of 30.58% compared to 311.27% at posto-level, indicate the linear regression's ability to make better predictions at more coarse administrative divisions. To avoid multicollinearity, in which independent variables in a model are correlated and negatively affect the reliability of the model's estimates (Alin, 2010), we excluded the standard deviation and variance features from the feature set.

The inspection of the random forest results revealed that the R^2 scores for all 4 models were nearly identical, ranging from 74.17% for the posto-level geographical

split model to 78.66% for the posto-level random split model, indicating consistent performance of the random forest model regardless of the administrative divisions and evaluation settings. The average R^2 score of the 4 random forest models (77.85%) was greater than that of the linear regression (73.16%), suggesting the former's superiority in the general task of population prediction. Since the random forest has the capability of modelling non-linear relationships between predictor and target variables, its expressive power is higher than that of a linear regression model. It must be noted, that a closer inspection of the target and predicted plots revealed that the random forest models underestimate the population counts for regions with high true population counts (B.2).

Lastly, the analysis of GCN and GAT/GraphSAGE models trained with empty graphs revealed a similar pattern to the linear regression model, where more accurate predictions on random split data were exhibited than on geographical splits. According to the R^2 score improvement of 2.82%, the GAT/GraphSAGE model outperformed the GCN on district-level predictions, whereas the GCN showed a 4.19% R^2 improvement over GAT on posto-level predictions. Both models tended to overestimate population counts in regions with low true population counts (B.3, B.4). Although the average R^2 score of the GCN model (78.07%) suggested its superiority over other baseline models, its error metrics were greater than those of linear regression and random forest, implying it explained variance but made larger errors. While GCN and GAT/GraphSAGE trained on empty graphs showed comparable performance in capturing relationships within population data, the higher error metrics compared to simpler models suggest the neural networks are struggling to learn the underlying pattern.

In summary, the evaluation of the baseline models revealed that the choice of the best model is highly dependent on the administrative level at which the predictions are made, the evaluation setting (random or geographical), and the evaluation metric. In general, the random forest model demonstrated the best performance overall, attributed to its ability to capture non-linear relationships. All baseline models showed more accurate predictions when assessed on random splits, struggling to generalize to distant regions, where the data might be differently distributed. Both neural network architectures—GCN and GAT/GraphSAGE trained on empty graphs- exhibited higher error metrics than their simpler comparative baselines, possibly struggling to develop a general pattern due to the small dataset size.

5.2.2 Geographical Boundary Graph GNN Results

		District-level					Posto-level				
		MAE	*MeAE*	*MAPE*, %	*MeAPE*, %	R^2, %	*MAE*	*MeAE*	*MAPE*, %	*MeAPE*, %	R^2, %
GCN, boundary	RS	105229.78	55232.30	55.16	41.78	17.50	52240.48	26432.74	210.53	65.39	20.00
	GS	113203.52	77620.11	77.95	47.40	-2.59	54998.75	25706.16	228.87	66.85	0.24
GAT, boundary	RS	104495.58	64170.23	63.23	41.78	25.68	53130.83	29769.73	278.30	66.36	35.48
	GS	113489.38	76416.77	72.86	47.46	5.39	55428.54	28250.00	575.65	70.38	13.37
GraphSAGE,	RS	54432.98	34595.47	39.71	22.36	**85.30**	29426.23	16001.46	277.83	41.70	82.67
boundary	GS	89925.97	56578.28	62.37	38.21	55.97	38880.40	19325.70	3677.19	50.66	66.84

Table 5.2: Cross-validation evaluation metrics for graph neural networks: GCN, GAT and GraphSAGE. All models were trained using the geographical boundary graph. Highlighted R^2 scores indicate the best-performing GNN across both graph types in the specific training setting.

Further, we discuss the results of the graph neural networks trained on graphs with edges constructed using geographical boundaries (G_1, G_2) (Table 5.2). Both the GCN and GAT architectures exhibit poor performance in predicting population at the posto and district levels, achieving R^2 scores in the range from -2.59% to 35.48%. No clear pattern emerges to distinguish which model of the two performs better, as the results vary across the administrative levels, evaluation settings, and performance metrics. In all the settings, the GCN and GAT models underperform the baseline models. For example, at the district-level prediction with random splits, the best-performing GCN model yields a *MAE* of 105229.78, which is more than double that of the best-performing baseline model at the particular setting -linear regression. These results suggest that the GCN and GAT models are ill-suited for population prediction under the conditions tested.

In contrast, the GraphSAGE model significantly outperforms the GCN and GAT architectures. When trained for district-level prediction and assessed on random splits, the model achieves an R^2 score of 85.30%, outperforming all baselines except linear regression. This finding indicates the utility of incorporating graph data in model training, as GraphSAGE trained using the boundary graph performs better than the baseline trained on the empty graph. For posto-level predictions evaluated on random splits, GraphSAGE demonstrates greater *MAE* and worse performance than most baselines. However, improvements in R^2, *MeAE*, and *MeAPE* metrics compared to the empty graph baseline suggest integrating graph data provides a small positive

impact on population prediction. On geographical splits, GraphSAGE underperforms all baselines, achieving a low R^2 score of 55.97% on district-level predictions and 66.84% on posto-level predictions.

Overall, the GCN and GAT architectures trained using a graph where edges are formed between geographically adjacent regions have proven unsuitable for population prediction, as evidenced by their high error metrics and low R^2 scores compared to the baselines. On the other hand, when assessed on random splits, the GraphSAGE model achieved results on par with the baselines, occasionally even outperforming them. Similarly to most baselines, the GNNs exhibit a significant performance drop when evaluated on geographical split cross-validation. This indicates the models' inability to generalise to unseen provinces, likely due to the high discrepancy between the provinces within training and testing sets.

5.2.3 Transportation Route Graph GNN Results

		District-level					Posto-level				
		MAE	*MeAE*	*MAPE*, %	*MeAPE*, %	R^2, %	*MAE*	*MeAE*	*MAPE*, %	*MeAPE*, %	R^2, %
GCN, transport	RS	103099.11	66133.38	57.35	41.78	27.17	49277.62	23411.62	297.76	63.61	36.60
	GS	113205.03	71233.23	68.24	48.40	1.89	50701.73	22662.5	1325.35	63.22	33.02
GAT, transport	RS	105901.75	67609.18	62.43	43.41	20.00	49851.58	24061.71	371.31	62.79	32.18
	GS	103862.14	64073.41	61.22	41.29	13.85	53359.38	27724.59	652.17	68.51	24.32
GraphSAGE, transport	RS	59440.60	41352.23	42.50	24.82	82.66	28634.20	16746.92	309.77	44.95	**87.56**
	GS	86394.63	57485.63	62.96	33.88	**62.07**	35946.72	22448.28	983.57	55.04	**81.20**

Table 5.3: Cross-validation evaluation metrics for graph neural networks: GCN, GAT and GraphSAGE. All models were trained using a transportation route graph. Highlighted R^2 scores indicate the best-performing GNN across both graph types in the specific training setting.

The results of the GNNs trained on graphs with edges constructed based on major transport connectivity routes (G_3, G_4) are illustrated in Table 5.3. Both the GCN and GAT models exhibit unsatisfactory performance, as indicated by the high *MAE* values and low R^2 scores ranging from 1.89% to 36.60%. The GCN model slightly outperforms the GAT model, with lower *MAE* and improved R^2 scores on district-level random splits (7.17% improvement), as well as posto-level random (4.42%) and geographical (8.70%) splits. Nonetheless, all evaluation metrics point to the significant underperformance of the GCN and GAT models compared to the baselines.

Similar to the geographical boundary model results, the GraphSAGE model consistently outperforms both GCN and GAT in every training setting. On district-level random splits, the GraphSAGE exceeds the performance of the GCN and GAT/GraphSAGE empty graph baselines by R^2 of 1.14%. For posto-level predictions evaluated on random splits, the model outperforms all baselines except for the random forest model, achieving an R^2 score of 87.56%. Performance improvements over the empty graph baselines indicate the utility of incorporating the graph data alongside the region features when training and evaluating in the random split setting.

For district-level predictions assessed using geographical splits, the GraphSAGE model underperforms compared to all baselines across all metrics, demonstrating a high *MAE* of 86394.63. Nonetheless, when trained for posto-level population prediction and assessed on geographical splits, the GraphSAGE exhibits contradictory results. It is outperformed by all baselines as shown by its higher 35946.72 *MAE* and 22448.2 *MeAE* values, yet it surpasses all baselines, according to improved R^2 scores, implying that while GraphSAGE makes larger errors on individual predictions, it captures the general data distribution trend more effectively than the baseline models.

In general, the GCN and GAT models trained using a graph with edges formed between areas linked by major transportation routes are unsuitable for population prediction, as demonstrated by their high *MAE* and low R^2 scores. The GraphSAGE model demonstrates its applicability to the population prediction task by outperforming the majority of the baseline models in the random split and posto-level geographical split settings. Similar to the baseline models and GNNs trained using the boundary graphs, the GNNs trained on transportation graphs perform worse when assessed on geographical splits compared to random splits.

5.2.4 Geographical Boundary and Transportation Route GNN Result Comparison

To gain a better understanding of the difference between the models trained using geographical boundary graphs G_1, G_2 (Table 5.2) and the models trained using transportation route graphs G_3, G_4 (Table 5.3), we compared the performance of these models for each combination of GNN model architecture, administrative level and cross-validation split setting. It is evident that in the majority of the settings, the model trained using the transportation graph outperformed the boundary graph model, with *MAE* improvement ranging from 792.03 to 9627.24 and R^2 improvements ranging from

6.10% to 32.78%. Nonetheless, the opposite was true for the GAT and GraphSAGE models intended for district-level prediction when assessed on random splits.

The best model GNN in the district-level random split setting was the GraphSAGE model trained using the geographical boundary graph (R^2=82.66%). In the district-level geographical split and posto-level random and geographical split settings, the GNN, achieving the best performance was the GraphSAGE model trained using the transportation route graph, reporting R^2 scores of 62.07%, 87.56% and 81.20%, respectively. Figure 5.1 illustrates the heatmaps of the true population of administrative units in Mozambique, alongside the best-performing baseline and GNN predictions both in random and geographical split settings.

The superiority of the models trained using graphs with edges formed based on transportation links can be attributed to several explanatory reasons. First, the vertices connected by the transportation edges model how the interaction occurs in the real world providing a more meaningful predictive signal than the boundary edges. Major transportation routes enable the transit of people and supplies, both of which have a direct impact on population patterns (Wang and Chen, 2018). In the case of the boundary edge graph, the edges connect vertices that convey little meaning to the task objective, resulting in vertex representations, which contain more noise than useful information.

As seen GNN classification tasks (Zhu et al., 2020), a decrease in graph homophily results in a decrease in the model performance, which matches the findings of our experiments as boundary graphs G_1, G_3 have a lower homophily rate than the respective transportation graphs G_2, G_4 (Table 4.1) and performed worse. Lastly, it must be noted that the transportation graph contains some isolated vertices, for which the aggregation step is excluded when calculating the updated vector representation. It might be the case that by relying just on the vertex feature representations themselves, the model is capable of making more accurate predictions than when taking into account the surrounding vertices.

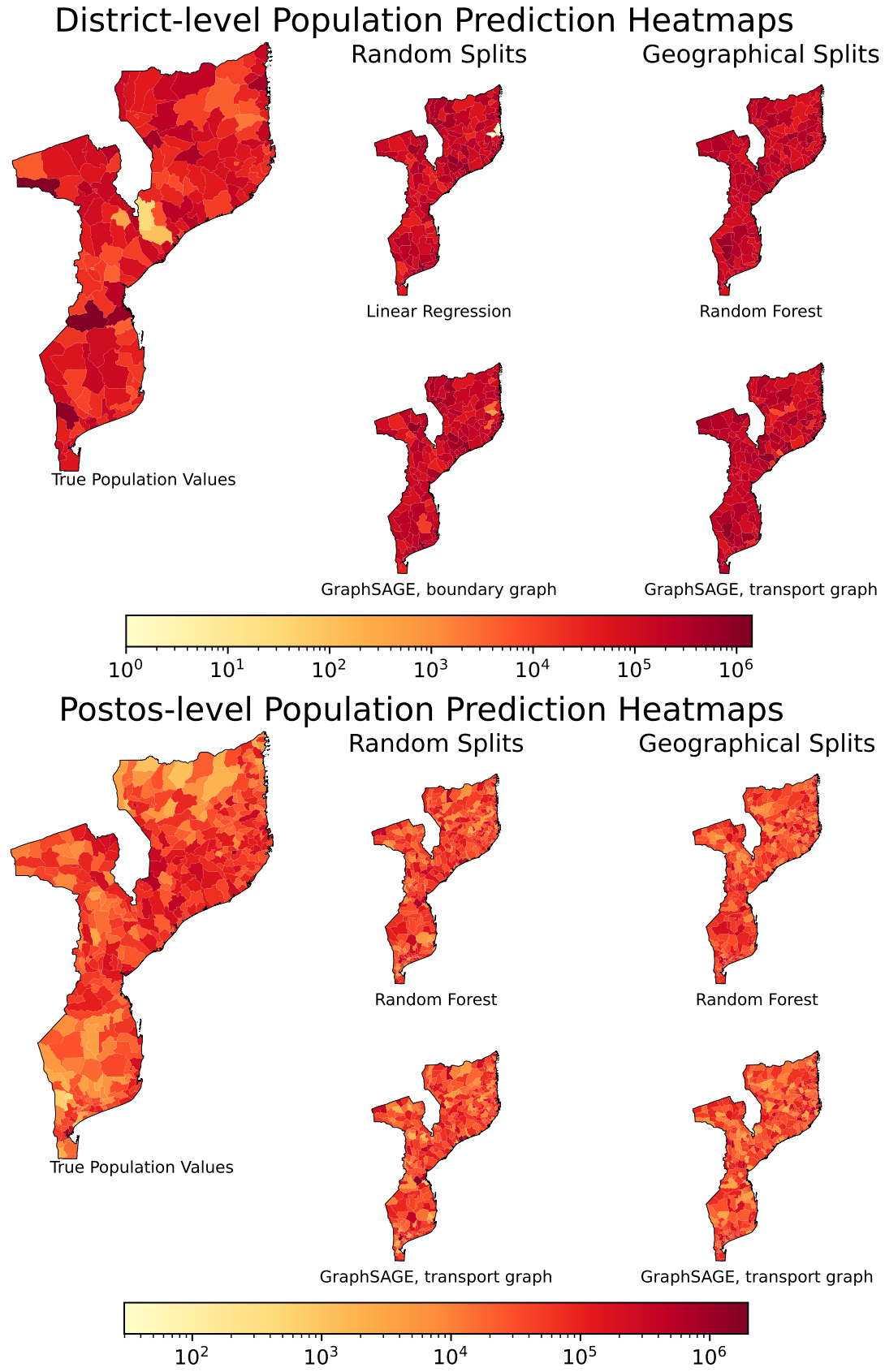

Figure 5.1: Heatmaps of Mozambique's true districts and postos population, alongside the predictions of the best baseline and GNN models.

5.3 Model Predictions and Confidence Intervals

To assess the proposed pipeline's predicted value trends and model uncertainty, we compare the best-performing GNN model's predictions and confidence intervals with the best-performing baselines in random and geographical split cross-validation settings. It must be noted that for the GNNs, the illustrated predictions represent the mean predictions from the Monte Carlo dropout, which we consider to be close enough approximates to the single-point estimates used in previously conducted model assessments. Figure 5.2 illustrates the district-level model predictions, their confidence intervals and target values. Equivalent behaviour is observed in posto-level prediction and confidence intervals, for which the reader is referred to Appendices B.11 and B.12.

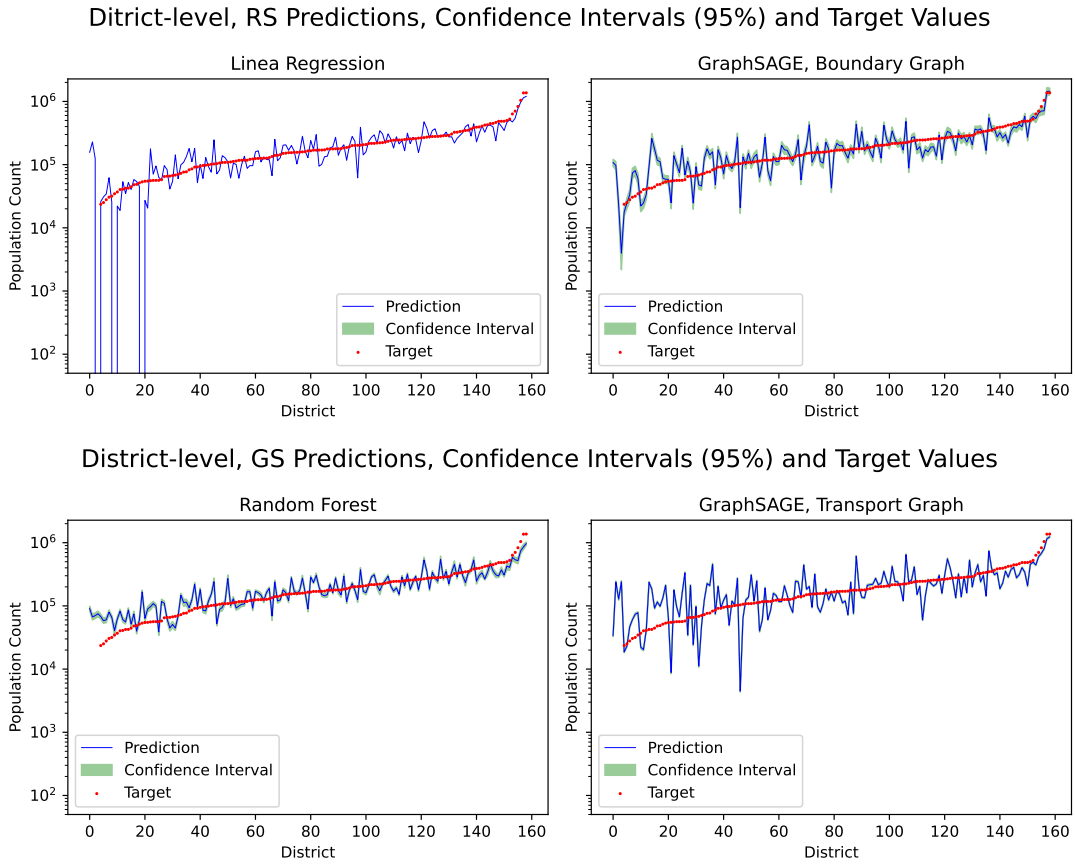

Figure 5.2: District-level model predictions and confidence intervals, alongside their targets in random split (RS) and geographical split (GS) settings. For each setting, we have illustrated the best-performing baseline and GNN model. The districts have been ordered in ascending order based on their target values.

The analysis of the best-performing random split models reveals that for low-population districts, the linear regression baseline significantly underestimates the citizen count, yielding negative predictions. The first half of the prediction line demon-

strates high-range fluctuations and a more tight fit as the population count in the target area increases, suggesting that the model is less suitable for predictions of districts with low population counts. Additionally, the wide confidence intervals of the linear regression model at the extreme ends of the target population indicate high uncertainty when making predictions in underpopulated and densely populated districts. This aligns with the poor performance at such districts, implying that the model recognises the prediction is likely to be incorrect.

The predictions of the GraphSAGE, transport graph model exhibit only slight underfitting at low-population districts compared to the linear regression model, making it a more suitable choice for underpopulated area prediction. Similar to the baseline, the population estimation gets more precise as the true citizen count in a district increases. Nonetheless, the overall model's prediction deviations from the target are higher than those of the linear regression. The GraphSAGE model's confidence intervals are approximately 2.5 times smaller, yet its predictions are less accurate than the linear regression ones, making it wrongly overconfident in its predictions.

In the case of model evaluation on geographical splits, the tight fit of the prediction line to the target values shows the random forest baseline's ability to generalize and predict in distant provinces. However, random forest slightly overestimates low-population districts and underestimates dense ones. The baseline's confidence intervals are narrower than the random split model's, making it highly confident in its predictions.

Lastly, the GraphSAGE, transport model assessed on geographical splits exhibits higher prediction deviations from the target value than its comparative baseline, with greater errors concentrated for population predictions of scarcely populated districts. GraphSAGE struggles to generalise to districts belonging to provinces absent from training data. Similarly, to the GraphSAGE model assessed on random splits, the model's narrow confidence intervals indicate overconfidence, as the model exhibits high certainty even for predictions which are far off from the target.

To summarise, the graph neural networks are outperformed by the linear regression and random forest baseline models in both random and geographical split cross-validation evaluation. While the GraphSAGE model demonstrates comparative performance to the linear regression model in the random split evaluation, its model makes greater errors in the geographical split evaluation, illustrating poor generalisation to distant regions (provinces). Random forest and GNN models exhibit high certainty in their predictions, with the GraphSAGE model often being wrongly overconfident.

5.4 Feature Importance

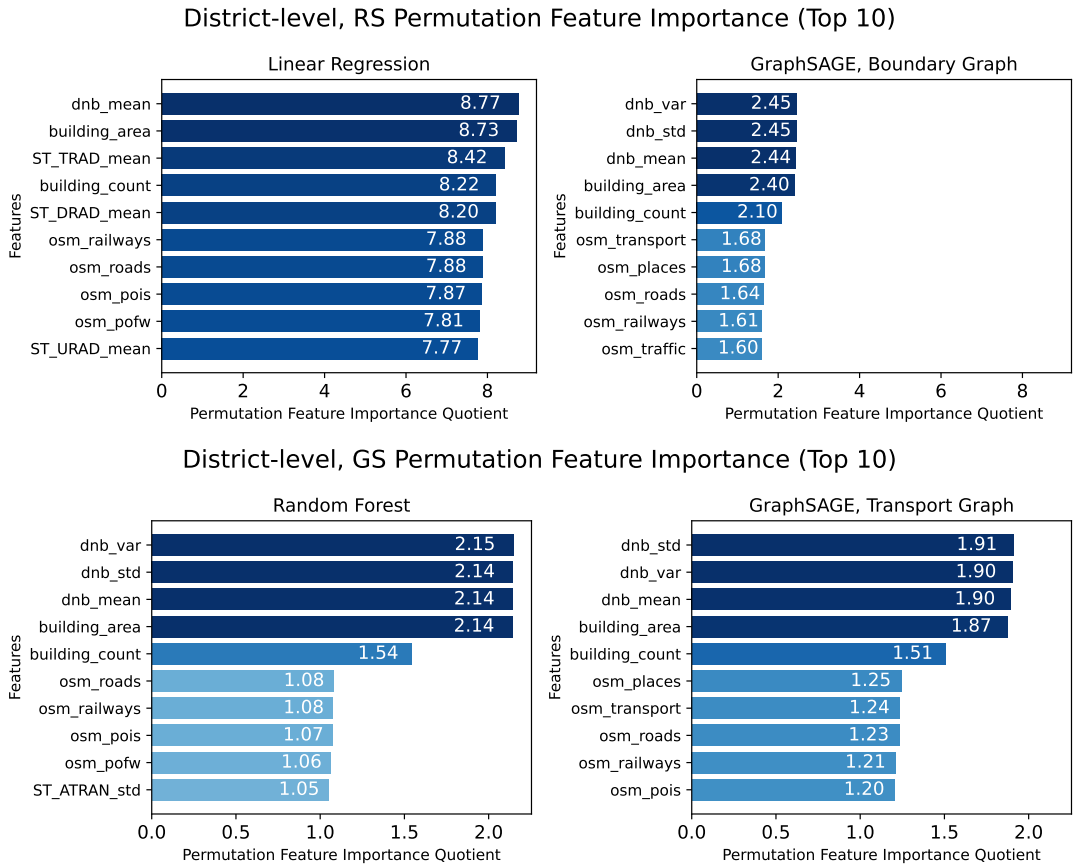

Figure 5.3: Top 10 features with highest permutation feature importance quotients for district-level prediction making in random split (RS) and geographical split (GS) settings. For each setting, the feature importance quotients are illustrated for the best-performing baseline and GNN models.

To provide greater interpretability of the trained model prediction-making decisions, we analyse which features have the highest influence on the model output. Figure 5.3 illustrates the 10 features with the highest permutation feature quotients of the best-performing baseline and GNN models in district-level population prediction in the random and geographical split cross-validation settings. Analogous trends are observed for posto-level prediction models, and interested readers can find the corresponding figures in Appendices B.7 and B.8.

All models illustrate nearly identical rankings of the most influential features, indicating that the features' strong predictive signal translates across multiple model architectures. For all models and training settings, the dominating features were the nighttime light mean, variance and standard deviation, which have proved to be an informative proxy for human presence measuring. Additionally, the building area

feature has been demonstrated to be nearly as important in population prediction making, ranking right after the nighttime light data and yielding nearly equal permutation feature importance scores. With a slight decrease in feature importance quotient, the majority of rankings are followed by the building count feature and several count features extracted from OSM maps.

It must be remembered that to avoid training the linear regression model on correlated data, the variance and standard deviation features were removed from the feature set, explaining their absence in the linear regression feature importance plots. Additionally, while the feature ranking of the linear regression model has remained nearly the same as the rest of the models, its feature importance quotients were nearly 4 times higher. This suggests a major increase in *MAE* when the features were permutated, likely due to the disruption in the linear relationship between the predictive and target variables.

5.5 GNN Results in the Context of Homophily

Further, we analyse the obtained GNN results with respect to their homophily rates and assess whether the findings in previous research on homophily in GNN apply to our vertex regression task. As already noted in Section 5.2.4, the models in the majority of cases when trained and evaluated in the same administrative and cross-validation split setting achieved greater performance when utilising transportation graphs rather than boundary graphs. Table 5.4 illustrates the *HReg* homophily rates of each graph used in each training setting, suggesting that in each scenario the homophily rate for the transportation graph was higher than that of the boundary graph. Apart from the 2 cases, highlighted in grey, we observe that the increase in homophily, results in an improvement in model performance, which aligns with the findings of Zhu et al. (2020) on the vertex classification task. Hence, it can be concluded that the *HReg* metric effectively captures the difference in model homophily rates, as showcased by consistent performance improvements in models trained on graphs with greater homophily.

As discussed in Sections 5.2.2 and 5.2.3, the GCN and GAT models significantly underperform the GraphSAGE models in all settings when trained on geographical boundary and transportation route edge graphs. Hence, our hypothesis of the high *HReg* homophily rate ensuring GCN and GAT model performances are on par with the GraphSAGE model has been proven wrong.

The poor GCN and GAT performances suggest that the incorporation of the neigh-

Model	Split	District-level		Posto-level	
		Boundary	Transportation	Boundary	Transportation
GCN	RS	0.8952	0.9094	0.9415	0.9511
	GS	0.8703	0.8871	0.9337	0.9511
GAT	RS	0.8952	0.9094	0.9415	0.9511
	GS	0.8763	0.8909	0.9415	0.9511
GraphSAGE	RS	0.8808	0.9002	0.9337	0.9443
	GS	0.8861	0.8974	0.9328	0.9443

Table 5.4: Comparison of boundary and transport graph *HReg* homophily values used for GNN model training in district and posto-level prediction, cross-validated on random splits (RS) and geographical splits (GS). The settings highlighted in grey, indicate the model where the boundary graph performance was superior to the transport graph model.

bouring region attributes hinders the model performance. The GCN and GAT architectures directly aggregate their vertex representation with the neighbouring ones, while the GraphSAGE preserves the vertex representation alongside the aggregated neighbour vector, allowing it to primarily rely on the vertex representation itself if the neighbouring vertex data provides more noise than useful signal.

The discrepancy between model results highlights the fact that the graphs used for training are of low homophily, as seen in the node classification task study by (Zhu et al., 2020). However, such a claim is contradictory to the graph homophily rates as measured by *HReg*, which indicates that all training graphs have high homophily, exceeding 0.87 *HReg* (Table 5.4). The finding can be attributed to the *HReg* metric's inability to accurately capture the true homophily rate of the graph in a regression setting, suggesting that the true homophily rate of boundary and transportation graphs is lower than approximated. Alternatively, it could be the case that the notion of GCN and GAT underperforming when trained on low homophily graphs only applies to classification tasks, but does not transfer to vertex regression tasks, implying that GCN and GAT models are unsuitable for regression tasks even though their homophily is high.

Chapter 6

Conclusion

Our study has introduced a graph neural network population prediction pipeline and tested its efficacy on 2nd (districts) and 3rd (postos) level administrative divisions of Mozambique, utilising a self-composed geospatial feature dataset. The pipeline was assessed using random cross-validation and geographical cross-validation, where the units were split based on their belonging to a 1st level administrative division (province), imitating a real-life scenario where the model must generalise to a remotely distant area with unknown population. The population prediction GNN pipeline consists of 3 modules: graph construction, feature extraction and predictive modelling.

We developed 2 types of graphs, where administrative units are vertices, and edges are formed either between units sharing a geographical boundary or units connected by a major transportation route. Our experiments show that GNNs trained with a transportation graph exhibit R^2 performance improvements from 6.10% to 32.78% over boundary graph models. We believe transportation graph edges are more informative, modelling real-life population migration trends.

We experimented with 3 GNN architectures: GCN, GAT and GraphSAGE. We found the GCN and GAT architectures to be unsuitable for the task of population prediction, achieving at best R^2 score of 36%. On the other hand, the GraphSAGE performance was on par with the baseline models, with best models reporting R^2 score of 85.30% on district-level prediction assessed on random splits and 62.07% on geographical splits, 87.56% on posto-level prediction assessed on random splits and 81.20% on geographical splits. While GraphSAGE preserves the feature vector of the area of interest along data from the neighbouring regions, the GCN and GAT models aggregate all that data into a single representation, resulting in reduced prediction accuracy when the connected vertices provide a minimal informative signal.

Our experiments have revealed that in the prediction-making of both administrative unit types, even the best-performing GNNs do not surpass the performance of the linear regression and random forest baselines. This is likely due to the small dataset size, which hinders the neural network's ability to learn the underlying pattern of the data. We note that both baselines and GNNs exhibit performance degradation when assessed on geographical split cross-validation compared to random splits, suggesting high data variability between the provinces in Mozambique. The confidence interval analysis of the linear regression showcases great uncertainty for incorrect predictions, whereas the random forest and GNN models tend to be overconfident, exhibiting high certainty even when making wrong population predictions.

Additionally, we explored whether observations from GNN classification tasks apply to vertex regression. Specifically, we investigated if GCN and GAT models, which typically underperform compared to the GraphSAGE model on low-homophily graphs, show similar trends in regression tasks. To measure the graph homophily, we utilised a recently proposed homophily measure $HReg$. Interestingly, although all graphs reported high homophily ($HReg > 0.87$), the performance of the GCN and GAT models was unsatisfactory. This suggests either that the $HReg$ metric fails to accurately capture the true homophily of the graph, or that the GCN and GAT architectures are inherently unsuitable for vertex regression tasks, regardless of the graph's homophily rate.

All models were trained using calculated features from self-composed remotely sensed data sources, including land use data, Landsat 8-9 imagery, OpenStreetMap extracts, building footprint and nighttime light data. Feature importance analysis revealed that the mean nighttime light value, total building area and count, as well as the count of OpenStreetMap points of interest, influence the model prediction the most.

6.1 Future Work

We discuss several future work possibilities by listing the limitations of our current work and suggesting improvements. First, the population counts used as target variables were not the true values but aggregate estimates. While we can utilise such data for model comparisons, the model predictions do not represent the true population in Mozambique for which model retraining on true census data would be required. Additionally, some predictor variables, including Landsat imagery and building data, were obtained in different at different times than the target population count. By acquiring predictor variables that match the state of the region exactly as it was when the population counts

were collected, the relationship between the remotely sensed features and population count could be modelled more accurately, achieving model performance improvements.

Further improvements could be made in the region feature calculation methodology. In our study, several features were obtained by aggregating pixel values of the respective region, causing significant information loss. A potential improvement could use an RNN to sequentially process pixel values and generate an abstract encoding for region features. Alternatively, splitting the region imagery into several image patches, and parsing each through a CNN, could obtain an abstract encoding for each patch. The encodings could then be averaged or parsed through an RNN, generating a feature vector for the whole region. The deep neural network architectures could potentially model more complex patterns with less data loss, improving overall model performance.

Future studies should investigate GNN performance when trained graphs with more vertices by using lower administrative division levels or creating artificial fine-grained regions. Neural networks require large training datasets to model the underlying data distribution (Sarker, 2021), whereas in our case the training sets were small. Alternatively, the training set size could be increased with data from another country. However, this would only apply to GAT and GraphSAGE models, as their inductive nature allows them to generalise to graphs of various structures, unlike the GCN model.

Since none of the explored GNN model performance surpassed that of the baselines, it is worth exploring alternative GNN training settings. By including edge features, such as the type of transport route, in the GNN training, the model could leverage additional information about the connections between the neighbours, by potentially giving higher importance to neighbours linked by certain types of routes. In addition, it is worth exploring other GNN architectures, such as H_2GCN, proposed by Zhu et al. (2020), due to their ability to make accurate predictions even for low homophily graphs, which we hypothesised to be the scenario for administrative level graphs in Mozambique.

The contradictory results of the high $HREg$ graph homophily rate but poor GCN and GAT performance indicate the need for further assessment of the interaction between homophily rate and vertex regression performance. Future studies must validate the $HReg$ measure on graphs from various domains and differing homophily rates. If the $HReg$ measure proves to be an accurate assessment of the graph homophily, studies exploring the effect on GCN and GAT model performance with various homophily rates on vertex regression tasks must be conducted to confirm the findings of our study.

Bibliography

Ahmed, N., Diptu, N. A., Shadhin, M. S. K., Jaki, M. A. F., Hasan, M. F., Islam, M. N., and Rahman, R. M. (2019). Artificial neural network and machine learning based methods for population estimation of rohingya refugees: Comparing data-driven and satellite image-driven approaches. *Vietnam Journal of Computer Science*, 06(04):439–455.

Alin, A. (2010). Multicollinearity. *WIREs Computational Statistics*, 2(3):370–374.

Balk, D., Leyk, S., Jones, B., Montgomery, M. R., and Clark, A. (2018). Understanding urbanization: A study of census and satellite-derived urban classes in the united states, 1990-2010. *PLoS One*, 13(12):e0208487. eCollection 2018.

Boo, G., Darin, E., Leasure, D. R., Dooley, C. A., Chamberlain, H. R., Lázár, A. N., Tschirhart, K., Sinai, C., Hoff, N. A., Fuller, T., Musene, K., Batumbo, A., Rimoin, A. W., and Tatem, A. J. (2022). High-resolution population estimation using household survey data and building footprints. *Nature Communications*, 13(1):1330.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Claire, I. (2007). Role of census data & population in disaster management. In *United Nations Sub-regional Workshop on Census Cartography and Management*, Bangkok,Thailand. RMSI Pvt. Ltd, India.

Diego Firmino Costa da Silva, J. P. E. and da Mota Silveira Neto, R. (2017). Urban and rural population growth in a spatial panel of municipalities. *Regional Studies*, 51(6):894–908.

Doda, S., Kahl, M., Ouan, K., Obadic, I., Wang, Y., Taubenböck, H., and Zhu, X. X. (2024). Interpretable deep learning for consistent large-scale urban population estimation using earth observation data. *International Journal of Applied Earth Observation and Geoinformation*, 128:103731.

Engstrom, R., Newhouse, D., and Soundararajan, V. (2020). Estimating small-area population density in sri lanka using surveys and geo-spatial data. *PLOS ONE*, 15(8):1–20.

Gadiaga, A. N., Bonnie, A. L., Lazar, A. N., Darin, E., and Tatem, A. J. (2023). Census disaggregated gridded population estimates for mozambique (2022), version 2.0.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning.

Georganos, S., Hafner, S., Kuffer, M., Linard, C., and Ban, Y. (2022). A census from heaven: Unraveling the potential of deep learning and earth observation for intra-urban population mapping in data scarce environments. *International Journal of Applied Earth Observation and Geoinformation*, 114:103013.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. *CoRR*, abs/1706.02216.

Hillson, R., Coates, A., Alejandre, J. D., Jacobsen, K. H., Ansumana, R., Bockarie, A. S., Bangura, U., Lamin, J. M., and Stenger, D. A. (2019). Estimating the size of urban populations using landsat images: a case study of bo, sierra leone, west africa. *International Journal of Health Geographics*, 18(1):16.

Jing Xia, Rui Li, X. L. G. L. and Peng, M. (2024). Scale effects-aware bottom-up population estimation using weakly supervised learning. *International Journal of Digital Earth*, 17(1):2341788.

Jones, M., Moeller, E. A., Meara, J. G., and Juran, S. (2021). The importance of geographic and demographic data from census for locating and mapping vulnerable populations. *Statistical Journal of the IAOS*, 37(1):13–17.

Karra, K., Kontgis, C., Statman-Weil, Z., Mazzariello, J. C., Mathis, M., and Brumby, S. P. (2021). Global land use / land cover with sentinel 2 and deep learning. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 4704–4707.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Leasure, D. R., Jochem, W. C., Weber, E. M., Seaman, V., and Tatem, A. J. (2020). National population mapping from sparse survey data: A hierarchical bayesian modeling framework to account for uncertainty. *Proceedings of the National Academy of Sciences*, 117(39):24173–24179.

Ma, L., Zhou, L., Blaschke, T., Yan, Z., He, W., Lu, H., Demuzere, M., Wang, X., Zhu, X., and Zhang, L. (2024). Projecting high resolution population distribution using local climate zones and multi-source big data. *Remote Sensing Applications: Society and Environment*, 33:101077.

Microsoft (2024). Global bing ml building footprints. `https://planetarycomputer.microsoft.com/api/data/v1/vector/collections/ms-buildings/tilesets/global-footprints/tiles/`. Accessed: 2024-06-05.

Mueller, T. T., Starck, S., Feiner, L. F., Bintsi, K.-M., Rueckert, D., and Kaissis, G. (2024). Extended graph assessment metrics for regression and weighted graphs. In Ahmadi, S.-A. and Pereira, S., editors, *Graphs in Biomedical Image Analysis, and Overlapped Cell on Tissue Dataset for Histopathology*, pages 14–26, Cham. Springer Nature Switzerland.

Neal, I., Seth, S., Watmough, G., and Diallo, M. S. (2022). Census-independent population estimation using representation learning. *Scientific Reports*, 12(1):5185.

Neal, I., Seth, S., Watmough, G. R., and Diallo, M. S. (2021). Towards sustainable census independent population estimation in mozambique. *ArXiv*, abs/2104.12696.

OCHA Regional Office for Southern and Eastern Africa (ROSEA) (2019). Mozambique - subnational administrative boundaries. `https://data.humdata.org/dataset/cod-ab-moz?`

O'Hare, W. P. (2019). *The Importance of Census Accuracy: Uses of Census Data*, pages 13–24. Springer International Publishing, Cham.

OpenStreetMap contributors (2017). Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`.

Pelletier, F. (2020). Census counts, undercounts and population estimates: The importance of data quality evaluation. Technical Paper 2, United Nations, Department of Economics and Social Affairs, Population Division.

Peng, Y., Liu, J., Zhang, T., and Li, X. (2021). The relationship between urban population density distribution and land use in guangzhou, china: A spatial spillover perspective. *International Journal of Environmental Research and Public Health*, 18(22).

Platonov, O., Kuznedelev, D., Babenko, A., and Prokhorenkova, L. (2024). Characterizing graph datasets for node classification: homophily-heterophily dichotomy and beyond. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *CoRR*, abs/1811.12808.

Robinson, C., Hohman, F., and Dilkina, B. (2017). A deep learning approach for population estimation from satellite imagery. New York, NY, USA. Association for Computing Machinery.

Román, M. O., Wang, Z., Sun, Q., Kalb, V., Miller, S. D., Molthan, A., Schultz, L., Bell, J., Stokes, E. C., Pandey, B., Seto, K. C., et al. (2018). Nasa's black marble nighttime lights product suite. *Remote Sensing of Environment*, 210:113–143.

Rusch, T. K., Bronstein, M. M., and Mishra, S. (2023). A survey on oversmoothing in graph neural networks.

Sarker, I. H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):420.

Seber, G. A. F. and Lee, A. J. (2003). *Linear Regression Analysis*. Wiley Series in Probability and Mathematical Statistics. Wiley, 2nd edition.

Sirko, W., Kashubin, S., Ritter, M., Annkah, A., Bouchareb, Y. S. E., Dauphin, Y. N., Keysers, D., Neumann, M., Cissé, M., and Quinn, J. (2021). Continental-scale building detection from high resolution satellite imagery. *CoRR*, abs/2107.12283.

Tong, Q. and Qiu, F. (2020). Population growth and land development: Investigating the bi-directional interactions. *Ecological Economics*, 169:106505.

UNFPA (2023). Mozambique population data 2023. `https://data.humdata.org/dataset/cod-ps-moz`.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks.

Wang, L. and Chen, L. (2018). The impact of new transportation modes on population distribution in jing-jin-ji region of china. *Scientific Data*, 5(1):170204.

Weber, E. M., Seaman, V. Y., Stewart, R. N., Bird, T. J., Tatem, A. J., McKee, J. J., Bhaduri, B. L., Moehl, J. J., and Reith, A. E. (2018). Census-independent population mapping in northern nigeria. *Remote Sensing of Environment*, 204:786–798.

Xiao Huang, Cuizhen Wang, Z. L. and Ning, H. (2021). A 100 m population grid in the conus by disaggregating census data with open-source microsoft building footprints. *Big Earth Data*, 5(1):112–133.

Yagoub, M., Tesfaldet, Y. T., AlSumaiti, T., Al Hosani, N., and Elmubarak, M. G. (2024). Estimating population density using open-access satellite images and geographic information system: Case of al ain city, uae. *Remote Sensing Applications: Society and Environment*, 33:101122.

Zhou, Y., Zheng, H., Huang, X., Hao, S., Li, D., and Zhao, J. (2022). Graph neural networks: Taxonomy, advances, and trends. *ACM Trans. Intell. Syst. Technol.*, 13(1).

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. (2020). Generalizing graph neural networks beyond homophily. *CoRR*, abs/2006.11468.

Appendix A

Selected Hyperparameter Values of Trained Models

A.1 Hyperparameters of the Random Forest Model

Model	Administrative level	Split	Node Splits
Random Forest	District-level	RS	2
		GS	2
	Posto-level	RS	10
		GS	5

Table A.1: Hyperparameter values of the random forest models in each administrative level and cross-validation split setting. The RS refers to the random split, whereas the GS refers to the geographical data split.

A.2 Hyperparameters of the Graph Neural Network Models

Model	Graph	Administrative Level	Split	Epochs	Hidden Layer Size	Message-Passing Layers	Learning Rate	Dropout Rate
GCN	Empty Graph	District-level	RS	519	16	2	0.1	0.2
			GS	589	16	4	0.01	0.1
		Posto-level	RS	169	8	3	0.1	0.05
			GS	256	8	2	0.1	0.2
	Boundary Graph	District-level	RS	487	64	2	0.01	0.25
			GS	51	128	7	0.1	0.15
		Posto-level	RS	114	16	2	0.1	0.15
			GS	211	4	6	0.1	0.2
	Transport Graph	District-level	RS	224	16	2	0.1	0.15
			GS	33	16	8	0.1	0
		Posto-level	RS	581	16	2	0.01	0.25
			GS	262	8	2	0.1	0.2
GAT	Empty Graph	District-level	RS	190	16	3	0.1	0.1
			GS	370	16	2	0.1	0.2
		Posto-level	RS	60	32	4	0.1	0.1
			GS	226	4	4	0.1	0
	Boundary Graph	District-level	RS	290	64	2	0.1	0.05
			GS	289	128	5	0.01	0.05
		Posto-level	RS	189	256	2	0.1	0.2
			GS	539	128	2	0.1	0.25
	Transport Graph	District-level	RS	436	32	2	0.1	0.15
			GS	557	32	5	0.01	0.25
		Posto-level	RS	291	16	2	0.1	0.15
			GS	95	64	2	0.1	0.05
Graph SAGE	Boundary Graph	District-level	RS	213	128	4	0.01	0.05
			GS	342	4	3	0.1	0.1
		Posto-level	RS	300	128	6	0.01	0.1
			GS	125	64	7	0.01	0.25
	Transport Graph	District-level	RS	245	128	3	0.01	0.1
			GS	85	32	4	0.1	0.1
		Posto-level	RS	238	32	4	0.01	0.1
			GS	379	8	4	0.01	0.15

Table A.2: Hyperparameter values of the random forest models in each administrative level and cross-validation split setting. The RS refers to the random split, whereas the GS refers to the geographical data split.

Appendix B

Model Performance Evaluation

B.1 Graphical Comparison of Target and Predicted Model Values

B.1.1 Baseline Models

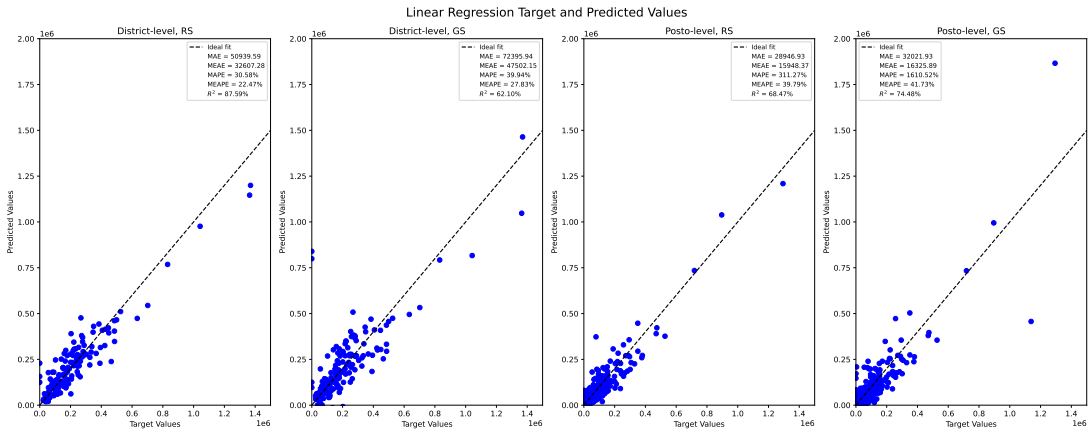

Figure B.1: Plots of target and predicted values of the four linear regression models. This includes results from both district-level and posto-level population prediction models, evaluated using random splits (RS) and geographical splits (GS). Model evaluation metrics are also presented for comparison.

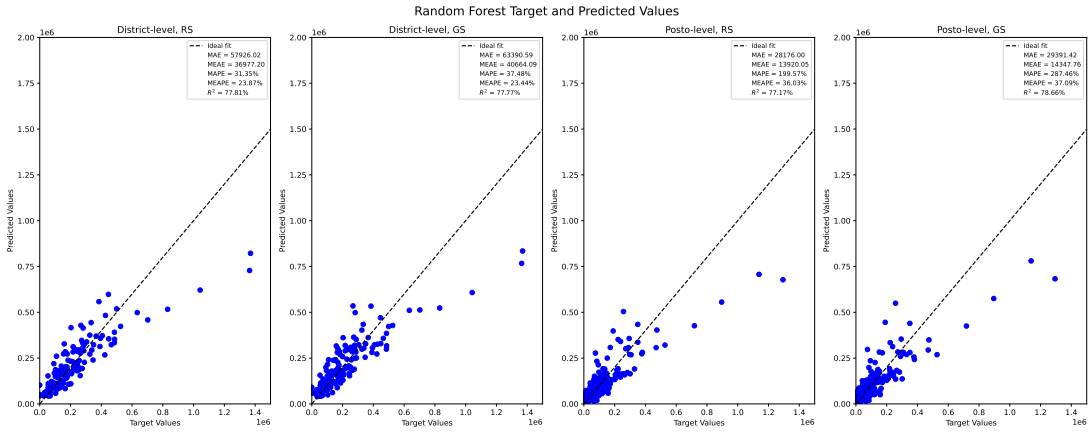

Figure B.2: Plots of target and predicted values of the four random forest models. This includes results from both district-level and posto-level population prediction models, evaluated using random splits (RS) and geographical splits (GS). Model evaluation metrics are also presented for comparison.

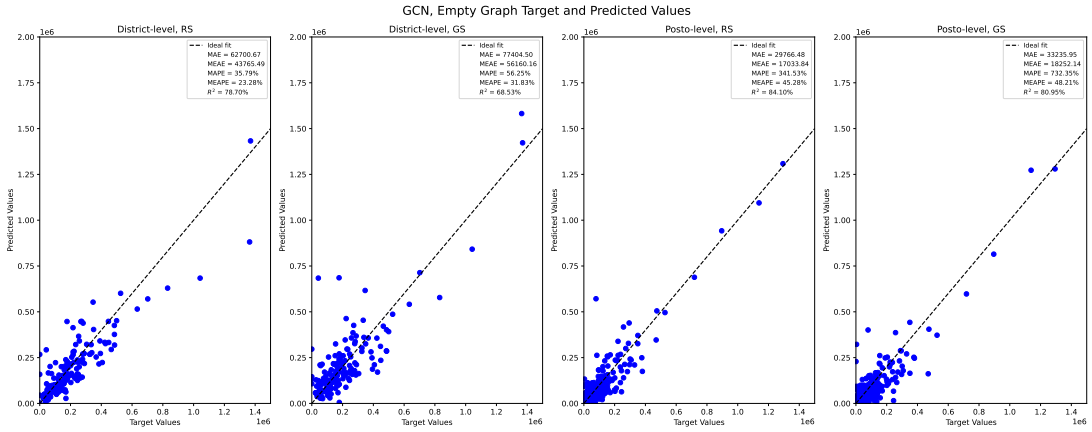

Figure B.3: Plots of target and predicted values of the four GCN models, trained using the empty graph. This includes results from both district-level and posto-level population prediction models, evaluated using random splits (RS) and geographical splits (GS). Model evaluation metrics are also presented for comparison.

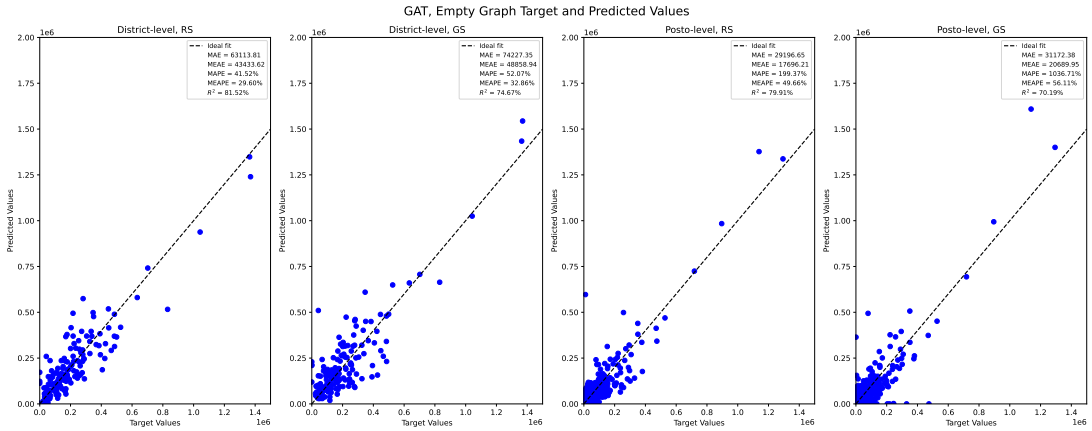

Figure B.4: Plots of target and predicted values of the four GAT models, trained using the empty graph. This includes results from both district-level and posto-level population prediction models, evaluated using random splits (RS) and geographical splits (GS). Model evaluation metrics are also presented for comparison.

B.2 Feature Importance

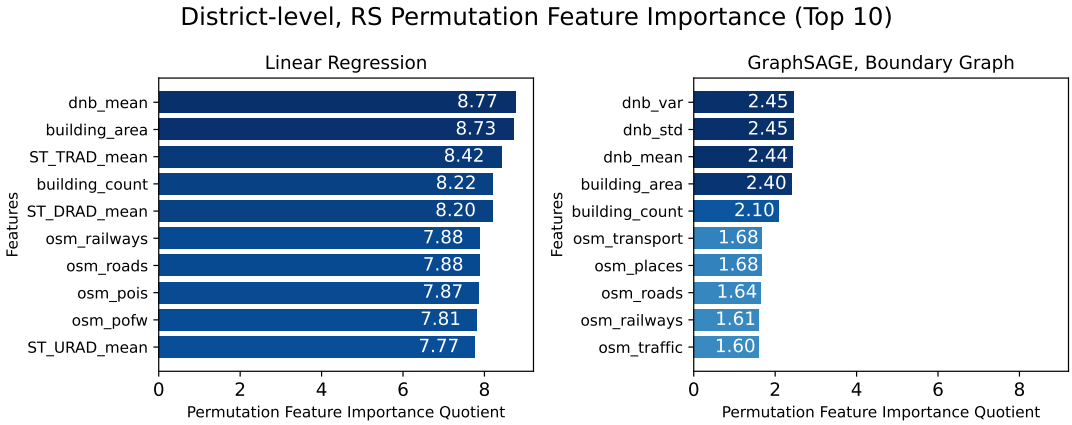

Figure B.5: Top 10 features with highest permutation feature importance quotients for district-level prediction in random split (RS) settings of the best-performing baseline and GNN model.

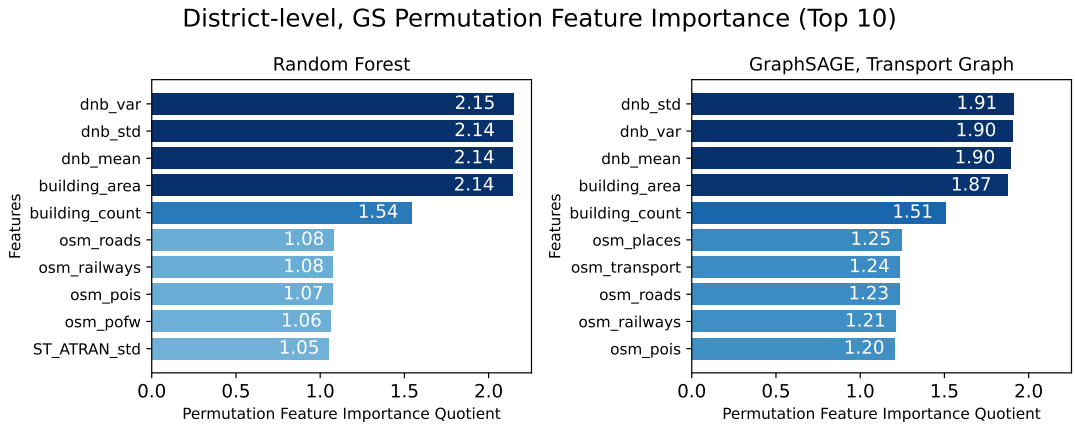

Figure B.6: Top 10 features with highest permutation feature importance quotients for district-level prediction in geographical split (GS) settings of the best-performing baseline and GNN model.

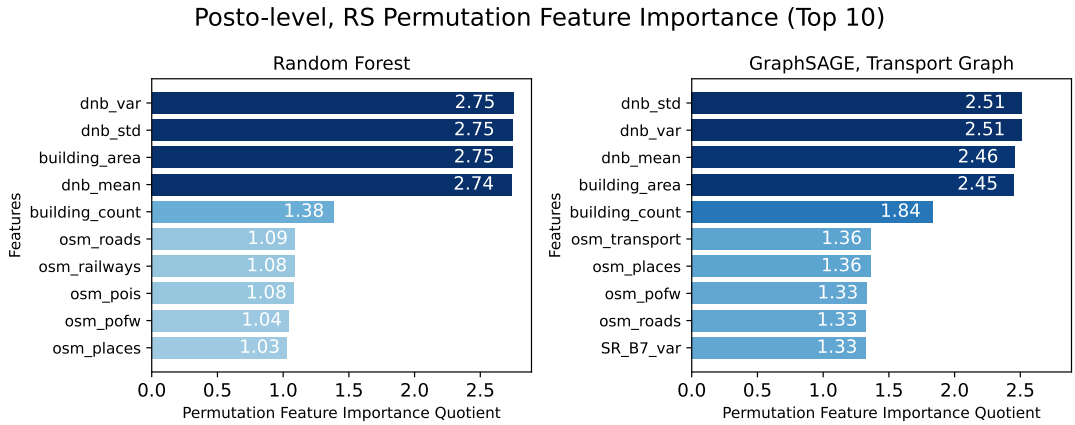

Figure B.7: Top 10 features with highest permutation feature importance quotients for posto-level prediction in random split (RS) settings of the best-performing baseline and GNN model.

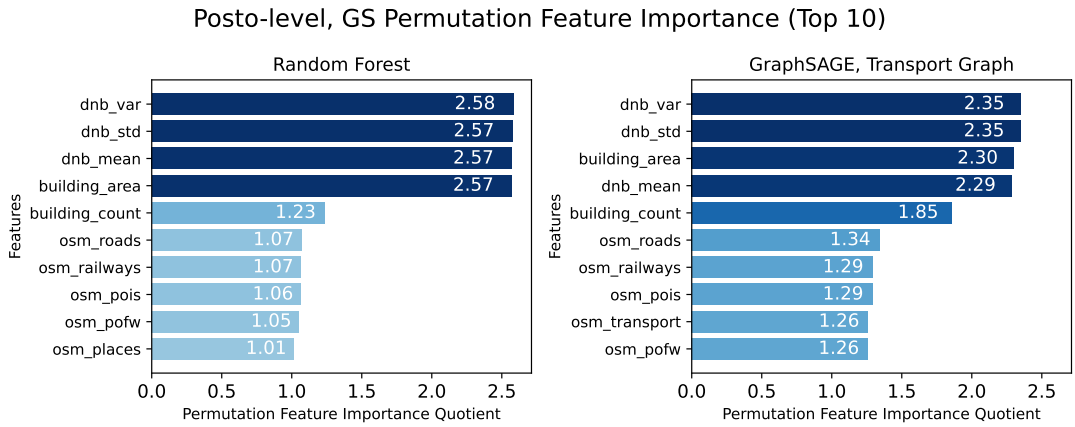

Figure B.8: Top 10 features with highest permutation feature importance quotients for posto-level prediction in geographical split (GS) settings of the best-performing baseline and GNN model.

B.3 Predictions and Confidence Intervals

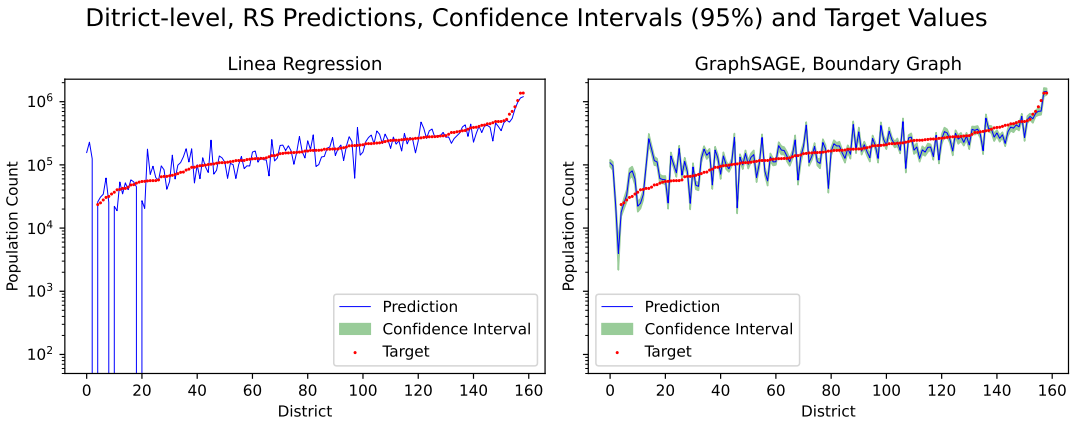

Figure B.9: District-level model predictions and confidence intervals, alongside their targets in random split (RS) setting. We have illustrated the best-performing baseline and GNN model. The districts have been ordered in ascending order based on their target values.

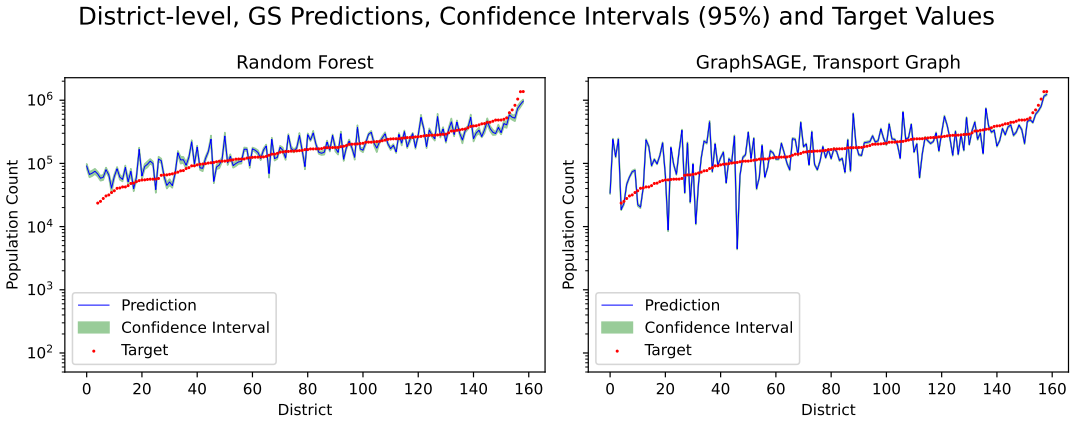

Figure B.10: District-level model predictions and confidence intervals, alongside their targets in geographical split (GS) setting. We have illustrated the best-performing baseline and GNN model. The districts have been ordered in ascending order based on their target values.

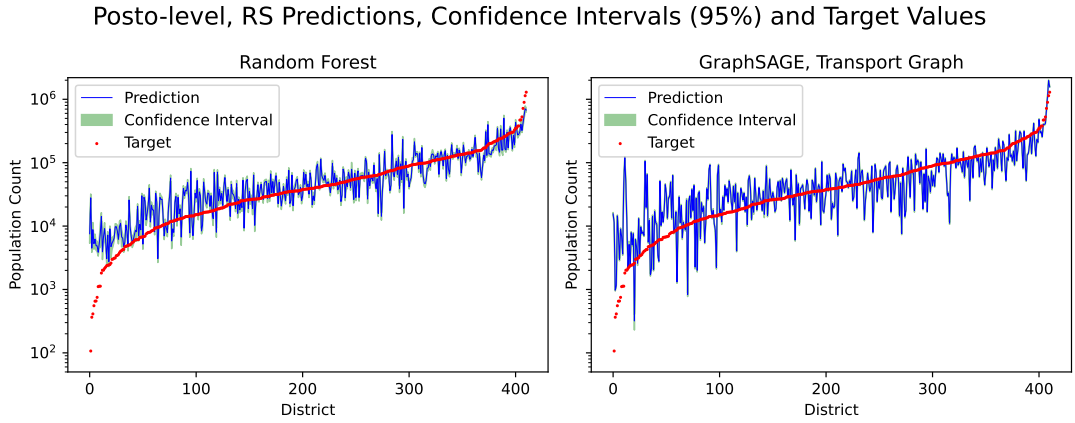

Figure B.11: Posto-level model predictions and confidence intervals, alongside their targets in random split (RS) setting. we have illustrated the best-performing baseline and GNN model. The postos have been ordered in ascending order based on their target values.

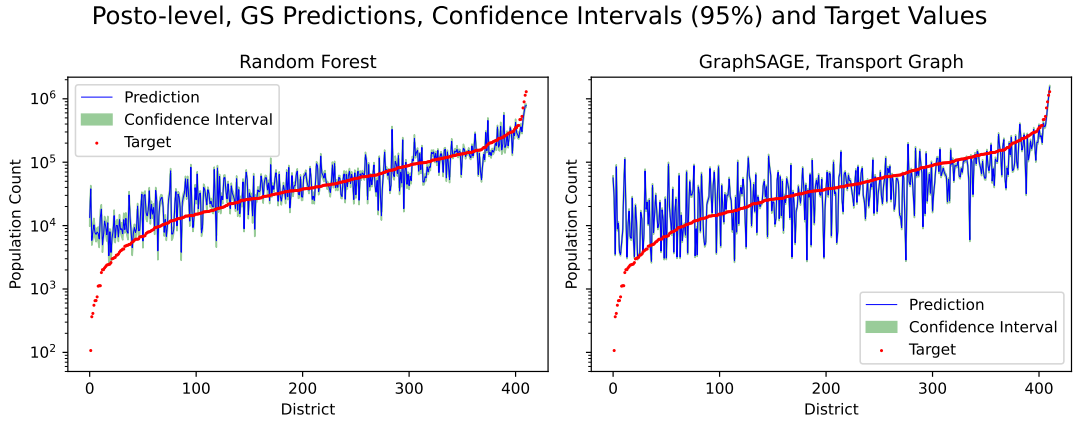

Figure B.12: Posto-level model predictions and confidence intervals, alongside their targets in geographical split (GS) setting. We have illustrated the best-performing baseline and GNN model. The postos have been ordered in ascending order based on their target values.