Hochschule Neubrandenburg
University of Applied Sciences

Computational Network Analysis; Statistics & Topologies of German Cities

Master of Engineering Geodesy and Geoinformatics

Rugilandavyi Barabona Mubondo
lg20028@hs-nb.de
2820004
Neubrandenburg, June 2021

Abstract

Cities exhibit a variety of street network patterns and configurations that shape human activities. This study models and analyzes the street networks of urban areas in Germany, using boundaries derived from the OpenStreetMap. Street network data are acquired and modeled from OpenStreetMap with the open-source OSMnx software package. In total, this study models thousands of OpenStreetMap street network nodes and edges across 6 urban areas of Germany. This paper presents the study's computational workflow, uses open data repositories of ready-to-use global street network models and calculates indicators, and discusses the results of the street network analysis of the different cities.

Acknowledgement

I would like to express special thanks to my supervisors of this project Prof. Andreas Wehrenpfennig and Prof. Ralf Löwner from the Hochschule Neubrandenburg University of Applied Sciences, also my sincere gratitude to Prof. Geoff Boeing, an Assistant Professor in the Department of Urban Planning and Spatial Analysis at University of Sothern California (USC) and the Director of USC's Urban Data Lab, for developing the essential OSMnx package and making it public together with his important research papers for my reference.

TABLE OF CONTENTS

1. INTRODUCTION

Street networks shape the city. They structure the circulation patterns of people and goods and underlie urban accessibility. Differences in street network geometry and topology reflect different cultures, political systems, urbanization eras, technology, design paradigms, climates, and terrain and more. These networks dictate how transportation is conducted within a city/area.

Comparative street network studies have been limited by data access and computational capabilities;

1.1. The Problem

Consistent urban area boundaries delineation and transport network mapping was difficult. Even if consistent study sites could be established, it was still very difficult to collect accurate and abundant street network data of just anywhere around the world. Also it was difficult to manipulate and organize the hundreds of millions of geospatial elements and model them in a graph-theoretic way, then compute their geometric and topological indicators.

1.2. The Objective

Spatial networks such as streets, paths, and transit lines organize the human dynamics of complex urban systems. They shape travel behavior, location decisions, and the texture of the urban fabric, therefore;

This study models the individual street networks of the urban areas specified, it calculates their geometric and topological indicators, and analyzes them individually and comparatively. Using OSMnx, it downloads and models urban-scale street network data from OpenStreetMap, then calculates indicators for each of the specified administrative boundaries of the urban area.

Measuring these street network patterns and their statistics can help researchers, planners, and community members understand local histories of urban design, transportation planning, and morphology; evaluate existing transportation system patterns and configurations; and explore new infrastructure proposals and alternatives. It also furthers the science of cities by providing a better understanding of urban patterns and how they correspond to evolutionary mechanisms, planning, and design.

2. METHODOLOGY

2.1. The Computational Environment

This study used several tools, technologies, and open data to
model the individual street networks. The tools included, Anaconda, which is the Graphical User Interface (GUI) and software for launching data science environments and manage conda packages easily together. PyQt Console for installing the OSMnx python package
Spyder, which is the Python programming environment for performing the network analysis and running the OSMnx package.
QGIS, the python-downloaded vector files of street networks graphs/maps were imported into QGIS, visualized and exported as image files.

2.2. Street Network Analysis with OSMnx

Firstly the necessary Python modules were imported, starting with matplotlib which is a package for data visualization and plotting. NetworkX is a package for generic network analysis. pprint pretty printed Python data structures and made them easier to read inline.

Then OSMnx package was imported, configured, and its version number as displayed:

The configuration step told OSMnx to log its actions to the terminal window and use a cache. The cache then saved a local copy of the data downloaded by OSMnx to prevent re-downloading the same data every time the code is run, hence configuration served to prevent data redundancy.

2.2.1. Graph Modeling

This study used OSMnx to download street network data from OpenStreetMap and created graph models of the specified urban area's drivable roads/street network. It repeated this for every specified administrative polygon of a places. It retrieved all public drivable streets including living streets, shared streets and such in the models, but excluded streets and other pathways where motor traffic was forbidden, for example, service roads, alleyways, parking lots and such non-drivable ways.

Spatial graphs usually have adverse periphery effects due to an artificial boundary being imposed, OSMnx therefore removed some of these effects by downloading and modeling a bit larger area than that requested to correctly calculate node degrees and later attenuated the peripheral nodes and edges that were outside the requested boundary polygon.

OSMnx then simplified the topology of the downloaded graphs and kept nodes only at the true intersections and dead ends, while also it preserved the true spatial geometry of all edges connecting the nodes. This was important in performing network analysis with OpenStreetMap data, such as calculating intersection density or average node degree. Simplification produced model that correspond better to graph theory and transportation cartography where the nodes represent intersections and dead-ends, and edges represent street segments.

The final downloaded street network models were saved as vector ESRI shapefiles together with all their attribute data, this made them workable with any GIS software environment, in this case QGIS.

2.2.2. Indicator Calculation

After acquiring all the models for the specified places, each saved ESRI shapefile was loaded by OSMnx to calculate each indicator described in Table 1. These indicators were merged and saved as a CSV-formatted file. The detailed descriptions serve to interpret the fields in these indicators' dataset.

The basic statistics were computed and inspected using a python function that calculated the basic descriptive geometric and topological statistics for the imported graph. For an unprojected lat-lng graph, the tolerance and graph units were in degrees and the circuity_dist was in 'gc', but because OSMnx produces a projected graph, therefore the tolerance and graph units were in meters and the circuity_dist was in 'eucledian'.

Parameters description;
road(network.MultiDiGraph) is the input graph of drivable roads
area(numeric) is the land area of the study site in square meters, it must be greater than 0. If none, will skip all density-based metrics.
clean_intersects(bool), if True, calculate consolidated intersections count and density if the area is provided by the function consolidate_intersections

tolerance(numeric) the tolerance value passed along if clean_intersects=True, see consolidate_intersections function documentation for details and usage
circuity_dist(string) is for calculating straight line distances for circuity measurement where ‘gc’ is used for lat-long networks and ‘eucledian’ is for projected networks.

Table 1. Fields in the indicators dataset.

Indicator name	Indicator Type	Description
circuity_avg	Decimal	edge_length_total divided by the sum of the great circle distances between the nodes of each edge
clean_intersection_count	Integer	Count of physical street intersections (after merging nodes within 10 meters geometrically)
clean_intersection_density_km	Decimal	clean_intersection_count divided by area in square kilometers
edge_density_km	Decimal	edge_length_total divided by area in square kilometers
edge_length_avg	Decimal	mean edge length in the graph, in meters
edge_length_total	Decimal	sum of all edge lengths in graph, in meters
intersection_count	Integer	Count of physical street intersections that is nodes with more than 1 physical street connected to them
intersection_density_km	Decimal	intersection_count divided by area in square kilometers
k_avg	Decimal	Average node degree (undirected)

m	Integer	Count or number of edges
n	Integer	Count or number of nodes
node_density_km	Decimal	n divided by area in square kilometers
self_loop_proportion	Decimal	Proportion of edges that are self-loops
street_density_km	Decimal	
street_length_avg	Decimal	Mean street segment length (undirected edges), meters
street_length_total	Decimal	Total street length (undirected edges), meters
street_segments_count	Integer	Count of street segments (undirected edges)
streets_per_node_avg	Decimal	Count of physical streets that connect to each node on average
streets_per_node_counts	Associative array	dictionary with keys of number of physical streets connecting to a node, and values of number of nodes with this count
streets_per_node_proportion	Associative array	dictionary, same as previous, but as a proportion of the total, rather than counts

2.2.3. Visualizing Street(edges) Centrality

Street/edges centrality measures how central edges are in a network and is defined as the reciprocal of the sum of the distance-weighted shortest paths between the edge and every other edge in the network.

We use the NetworkX (OSMnx) it to calculate and visualize the closeness centrality of different streets in the network

First, we convert our graph to its line graph which inverts its topological definitions, meaning that the streets become nodes and intersections become edges. Then we calculate the closeness centrality of each node (i.e., street in the line graph):

After calculating the centrality of each street in the network, we visualize it with matplotlib via OSMnx's plot_graph function, using the inferno color map to represent streets centrality in bright yellow for most central streets and in dark purple for the least central streets

2.2.4. Visualizing Intersections(nodes) Centrality

Street/edges centrality measures how central edges are in a network and is defined as the reciprocal of the sum of the distance-weighted shortest paths between the edge and every other edge in the network

2.2.5. Street Network Orientation by Polar Histogram

Among other analyses, we can use OSMnx to calculate and visualize street network orientation. That is, what are the bearings and spatial orientations of the streets in the network.

OSMnx automatically calculates all of the streets' bearings. It calculates the compass bearing from each directed edge's origin node *u* to its destination node *v*. Then we visualize these bearings, binned together as a histogram to understand the relative frequency of the streets' spatial orientations.

We then project that histogram as a polar plot to match the compass bearings for a better visual comprehension of the network orientation.

2.2.6. Square-Mile Street Network

Drawing these cities at the same scale provides a revealing spatial objectivity in visually comparing their street networks and urban forms. We recreate these visualizations automatically with Python and the OSMnx package.

These visualizations reveal the difference and or similarity of different places at the same scale.

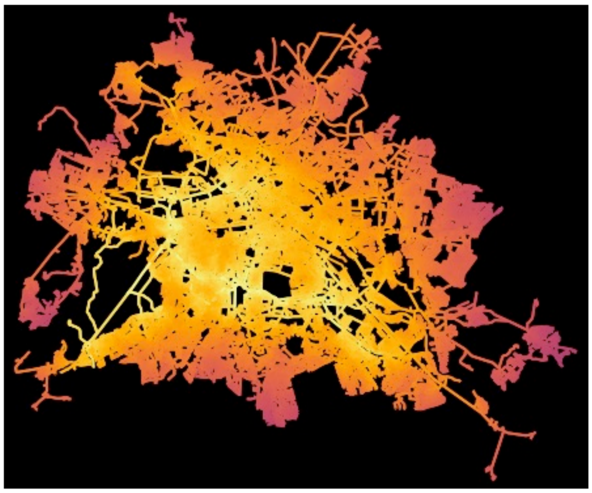
3. RESULTS & DISCUSSSION

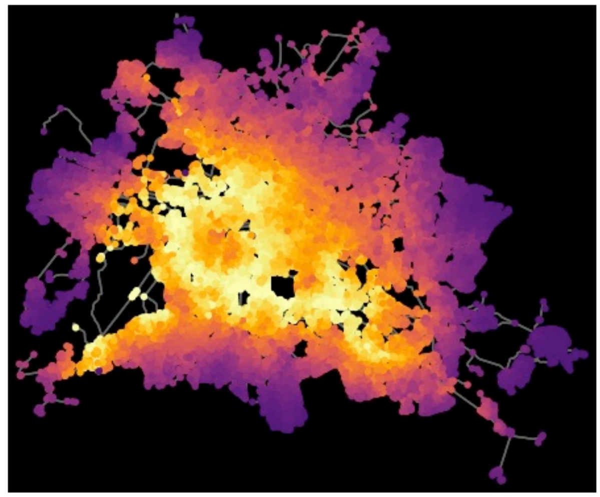
3.1. Basic Network Statistics

City or Town	Neubrandenburg	Berlin	Munich	Frankfurt	Cologne	Hamburg
Circuity average	1.09	1.04	1.04	1.06	1.05	1.08
Clean Intersections	869	19514	10134	5879	9660	14283
Intersections density	16.82	17.51	28.37	22.35	23.43	15.62
Edges density in km	8294	9554	13012	9273	10431	8246
Edges length average	125	146	128	122	120	144
Edges length total in m	428338	10650342	4648612	2439584	4301005	7540553
Intersection count	1134	24637	12625	8252	13242	18658
Intersection density	22.20	21.96	35.34	31.37	32.12	20.40
Average node degree, k	4.71	22.10	5.17	4.25	4.42	4.83
Number of edges, m	3423	72902	36306	20014	35922	52278
Number of nodes, n	1451	27951	14042	9415	16254	21622
Node density in km	28.09	25.07	39.39	35.79	39.42	23.69
Self-loop proportion	0.015	0.002	0.004	0.005	0.004	0.025
Street density in km	4587	5496	7655	6372	6469	4707
Street length average	123	145	130	122	119	138
Street length total	236899	6126574	2734890	1676262	2667358	4304382
Street segments	1933	42235	21041	13777	22322	31120
Streets per node avg	2.67	3.02	3.00	2.93	2.75	2.87

3.2. Edges and Nodes Centrality Visualizations

Berlin

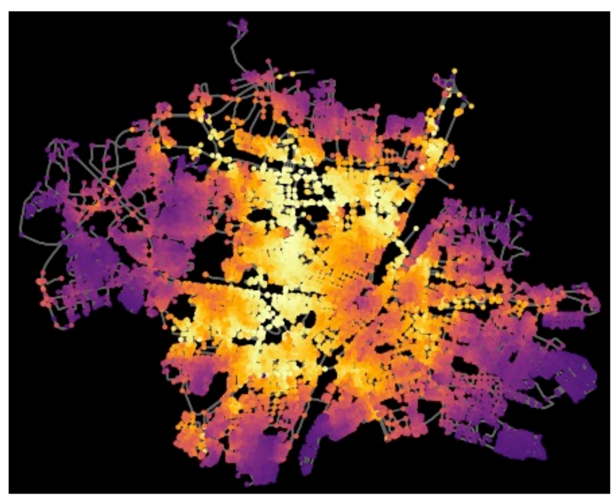




Edges Centrality

Nodes Centrality

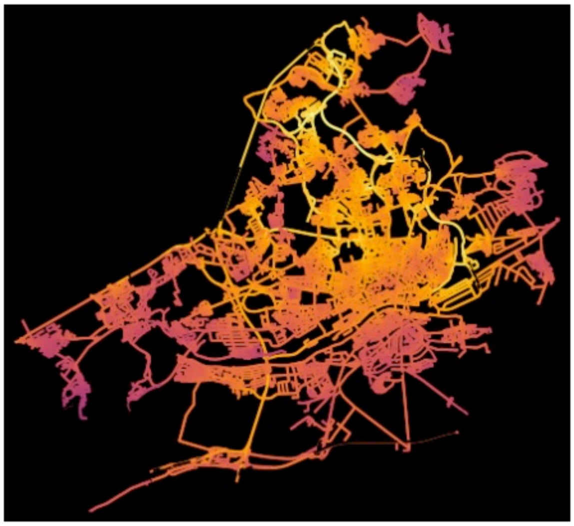
Munich



Edges Centrality

Nodes Centrality

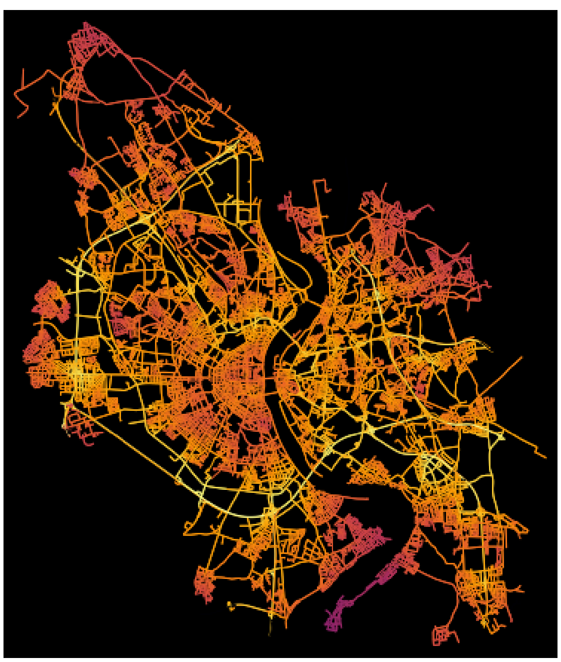
Frankfurt



Edges(Streets) Centrality

Nodes(Intersections) Centrality

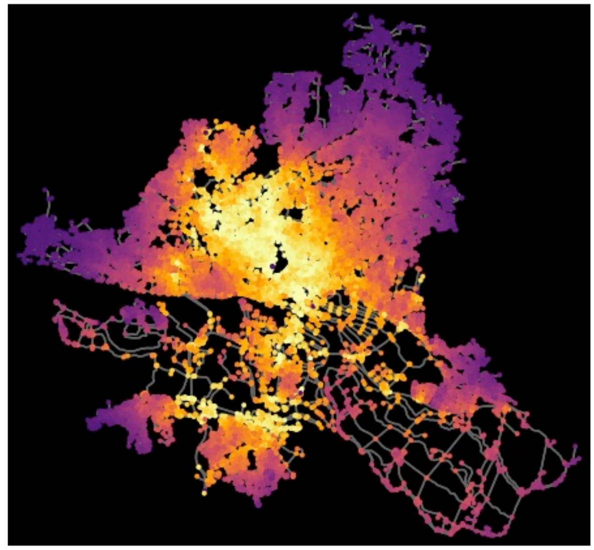
Cologne



Edges(Streets) Centrality

Nodes(Intersections) Centrality

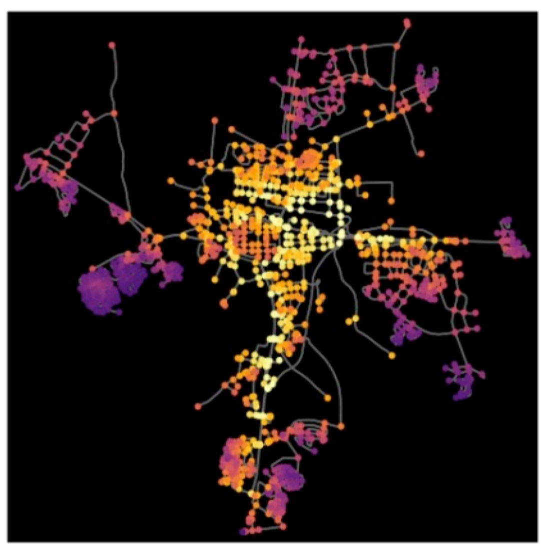
Hamburg



Edges(Streets) Centrality

Nodes(Intersections) Centrality

Neubrandenburg

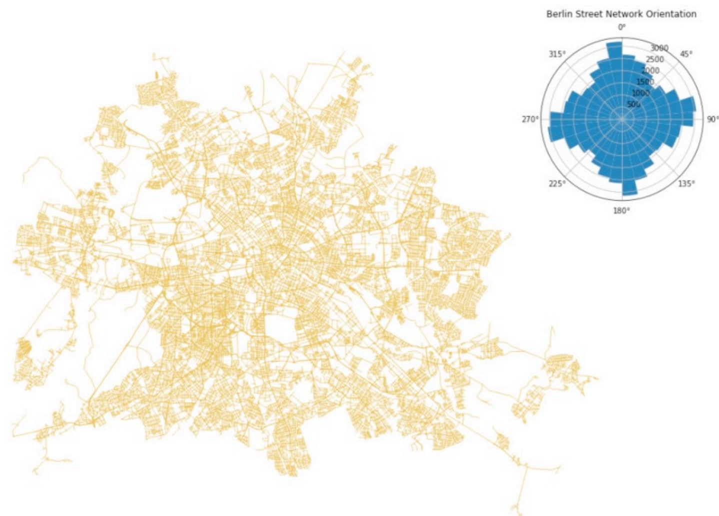


Edges(Street) Centrality Visualization

Nodes(Intersections) Centrality Visualization

3.3. Street Network Graph Models

Berlin

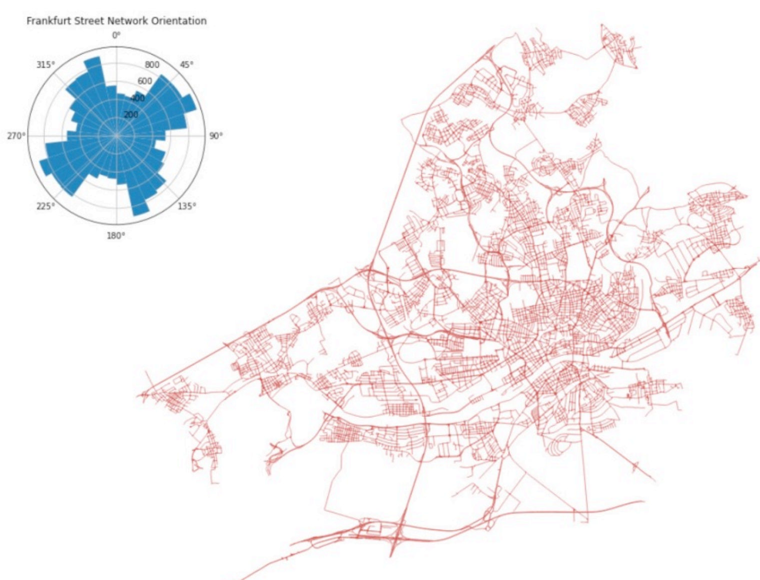


Berlin Street Network and Polar histogram for orintation

Munich

Munich Street Network and Polar histogram for orientation

Frankfurt

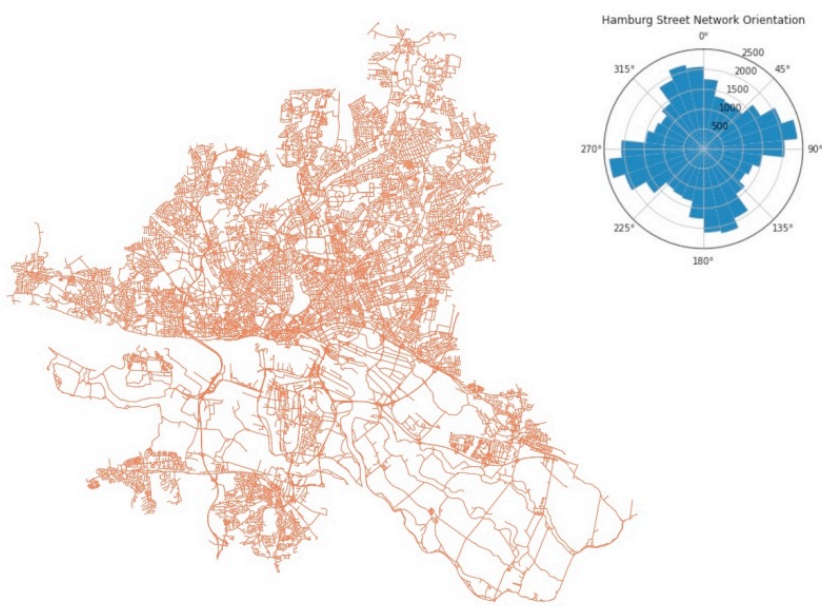


Frankfurt Street Network and Polar Histogram for Orientation

Cologne

Cologne Street Network and Polar Histogram for Orientation

Hamburg



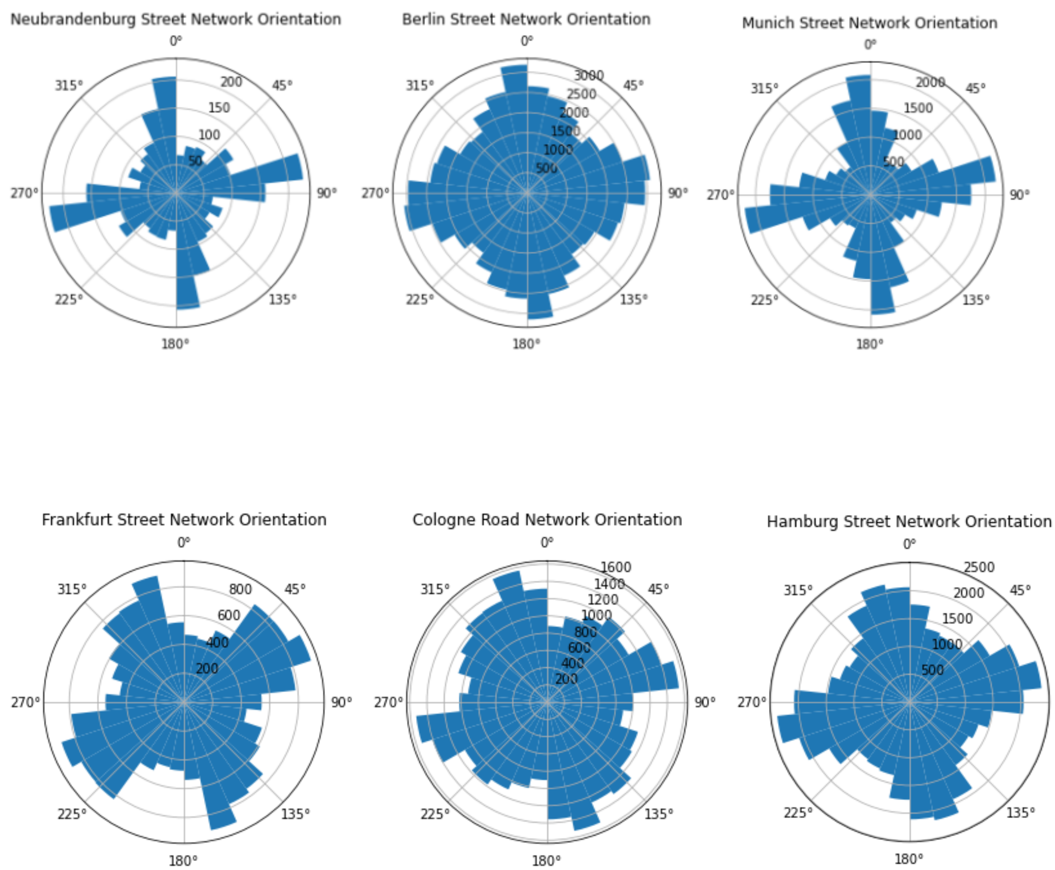
Hamburg Street Network and its Polar Histogram for Orientation

Neubrandenburg

Neubrandenburg Street Network and Polar Histogram for orientation

3.4. Street Network Orientations

COMPARISON OF STREET NETWORK ORIENTATIONS



Each of the cities above is represented by a polar histogram (aka rose diagram) depicting how its streets orient. Each bar's *direction* represents the compass bearings of the streets (in that histogram bin) and its *length* represents the relative frequency of streets with those bearings.

For Neubrandenburg and Munich have an at least approximate north-south-east-west orientation trend (i.e., 0°-90°-180°-270° are their most common four street bearing bins), but for the rest of the cities, Berlin, Frankfurt, Cologne and Hamburg they have the adjacent orientations (i.e., 10°-100°-190°-280° or 80°-170°-260°-350°) as their most common, the streets/roads in these cities are going outwards more towards all directions. Thus, even cities without a strong grid orientation often still demonstrate an overall tendency favoring north-south-east-west orientation e.g. as seen especially in Berlin and Cologne.

This study measures the entropy (or disordered-ness) of street bearings in each street network, along with each city's typical street segment length, average circuity, average node degree, and the network's proportions of four-way intersections and dead-ends. This shows that the entropy of cities of highest to lowest are Berlin, Cologne, Frankfurt, Hamburg then Munich and Neubrandenburg.

Entropy correlates strongly with the network topology too, but because the orthogonal grids of cities especially bigger cities like Berlin are oriented towards all directions they are well-arranged but exhibit higher entropy.

3.5. Figure Ground Diagrams

COMPARISON OF SQUARE-MILE GROUND FIGURES

Figure-ground diagrams shows the relationship between built and unbuilt/street space, in this study they were sampled from the center of the network graph. At the center of the Neubrandenburg square mile lies the Friedrich-Engels-Ring, from which Demminer, Woldegker, Neustrelitzer and Rostocker streets radiate outward to form more suburban topology.

Berlin has a more defined grid; it has a grid structure with collector roads parallel to arterials. while Munich, Frankfurt, Cologne have organic and irregular grids with discontinuous collector roads running parallel to the arterials.

4. CONCLUSION AND RECOMMENDATION

This study develops the tools necessary to perform network analysis of the roads of different places, but it can also be used to perform the same analyses with different types of transport infrastructure, such as railway, water ways, and all infrastructure that can be represented as vector lines from the OpenStreetMap. The network analysis and data of any location in the world can be performed and acquired respectively if it is available on OpenStreetMap.

OpenStreetMap provides incredibly valuable raw data, this study transforms that data into ready-to-use visual models and indicators through substantial processing. The topologically simplified graphs provide models that correspond much better to graph theory and transportation geography than raw OpenStreetMap data do, and they are much faster to run graph algorithms on because most such algorithms scale with node count. We could include elevation and grade data from Google Maps and SRTM that is often ignored in street network analytics.

These models to simulate trips, assess network vulnerability to flooding and sea level rise, or measure accessibility to points of interest. Therefore, this study is relevant in the fields of urban planning, transport engineering, risk analysis, suitability studies and many others that require geospatial data of this nature.

References and Data Sources

https://osmnx.readthedocs.io/en/stable/

https://github.com/gboeing/osmnx-examples/

https://www.openstreetmap.org/

Boeing, G. 2020. "Urban Street Network Analysis in a Computational Notebook.

Boeing, G. 2019. "Urban Spatial Order: Street Network Orientation, Configuration, and Entropy.

Boeing, G. 2017. "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks.