Department of
Statistics
Winter term 2023/24

Learning graph structures through graph representation learning

Mentor:
Christopher Caratiola

Submitted on:
April 30, 2024

Submitted by:

Bindushree Heggere Dharanendra Kumar (1677020) s4bihegg@uni-trier.de

Nagashree Arun Mysuru (1676623) s4naarun@uni-trier.de

Naveen Malla (1619019) s4namall@uni-trier.de

Thuy Quynh Nguyen (1678590) s4tqnguy@uni-trier.de

Table of Contents

List of Figures

1.Introduction

Street networks are the real arteries of urban areas, which functionally determine the structure and vitality of cities. They adopt the role of travellers who transport people and goods, and hence are the most important entities that shape the social, economic, and physical environments of cities. Knowing a street network's history and its current role in urbanization is indispensable to get a clear picture of the multi-faceted urban development process. With cities being recognized as the fundamental entities in environmental, economic, and social matters, street networks have become the subject of numerous multidisciplinary studies on transport, urban planning, geography, and physics over the past 50 years.

In order to identify the spatial structures of street networks, researchers have extensively modelled the empirical features and constructed models capable of the reproducing of their structural and topological properties. The representation of street networks as graphs, with intersections as vertices and segments as edges, led to the introduction and use of methods from network science and complexity science. Among other things, this approach has allowed scientists to study different properties of street networks, including their structural, topological, hierarchical and fractal properties, and to model their evolutionary paths.

In recent years, the accessibility of large open datasets, usually crowdsourced community data such as OpenStreetMap, and improvements in machine learning have transformed the way researchers study street networks. These breakthroughs are giving rise to a new era of parties focused on capturing the complexity of urban forms by combining traditional methods with data-driven approaches. The common approach to street network analysis, which uses manually created rules to extract key attributes, such as the degree of connectivity with link statistics or vertex centrality, may not be the most rational solution. However, large open databases with machine learning techniques allow the obtained street network pattern to be replaced by latent vector features known as embeddings. These new approaches to representational learning have shown a high degree of effectiveness compared to the previous approaches in the different applications.

The objective of this research was to develop a system capable of generating artificial street networks based on the methodology proposed by (Neira and Murcio et al., 2022). To this end, the study employed the techniques outlined by the authors in the data gathering, preprocessing, and implementation of the 0node model, which was designed to learn and estimate a distribution over a sequence of nodes. First, the spatial coordinates of the nodes are encoded in a suitable format that can be given as input to the transformer model, since a transformer can capture complex spatial linkages and dependencies in the data. In this study, the input format is a sequence of coordinate values (C^{seq}). As a sequence-to-sequence model, the transformer interprets the input sequence and uses an auto-regressive method to predict the output sequence ($C^{seq'}$).

Finally, this research utilised the node2vec model, an alternative approach to the Variational Graph Auto Encoder as described by (Neira and Murcio et al., 2022), a highly effective graph representation learning method, to study the analysis of street networks. Word2vec is replaced by node2vec to extract not only the semantic features of the network, but also its structure. This allows the generation of low-dimensional embeddings. The objective is to utilize node2vec to learn a distributed representation of the street network that encompasses both its spatial layout and local features. This approach emphasizes current work in the fields of street network analysis and graph representation learning. The model was trained on 35,974 different cities, resulting in the learning of representations of their street networks. Furthermore, the efficiency of the model was demonstrated by calculating metrics of the generated network, including circulation, average edges per node, average form factor and average compactness.

2. Literature review

The discovery and search for models that can quantitatively reproduce and replicate the properties of street patterns has been aided by a number of studies conducted over the years. In this sense, an overview of different street network models is provided by (Marshall et al., 2018), where street segments represent the edges and intersections serve as the vertices of the graph. These techniques have been used to improve the understanding of the structural, topological, hierarchical and fractal properties of street networks (Strano et al., 2012; Louf and Barthelemy et al., 2014; Arcaute et al., 2016; Murcio et al., 2015).

Recently, with the wide availability of new large-scale open datasets, such as crowdsourced volunteer geographic information, in particular OpenStreetMap (OSM contributors et al., 2017), and further advances in machine learning research, many studies have attempted to gain a more comprehensive understanding of the complexity underlying urban structures. Traditional street network tracing has been limited by manual, user-defined heuristics for extracting representational features, such as degree measures or centrality measures. Conversely, the use of machine learning approaches with large databases has enabled the identification of the topological structure of street networks through the visualisation of low-dimensional latent feature vectors. An advantage of this approach is that it does not require manual feature engineering and has been shown to outperform traditional methods in several domains. In addition, generative models using synthetic street network data as input have been shown to be useful for street network construction. VAEs trained on street network images were used by (Kempinska and Murcio et al., 2019), however, due to the low resolution of the images, the problem was not solved as finer elements of local streets were not captured and preserved.

(Hartmann et al., 2017) proposed that networks of streets could be created using a technique called Generative Adversarial Networks (GANs), which were able to reproduce general patterns while missing certain topological features. Although these models have been shown to reflect the general trends of the street networks well, the derived latent space does not preserve the properties of the input data set. More work needs to be done on how these latent spaces relate to the established street network. Converting the graphs from images back to graphs may also have an additional error in the results. (Neira and Murcio et al., 2022) aim to address some of the shortcomings of learning low-dimensional vector representations of street networks, which can be used in tasks such as street network morphology classification of different cities and road network generation. However, the goal of VGAE is to obtain a decoded adjacency matrix that is the reconstructed one. This research didn't yield a way to implement the input file. So, to tackle this challenge, an alternative solution was explored to this problem by using the node2vec model, which produces node embeddings that represents the connections between nodes.

The node2vec model brings out the idea of a special parallelism for graphs, learning the embeddings through efficient traversing of neighbourhood structures. The model may overcome the limitations of graph embedding to cover the spatial attributes that are needed to analyse the topological features of street networks. For example, (Grover and Leskovec et al., 2016) demonstrated the effectiveness of the node2vec algorithm in learning low-dimensional representations of nodes in street networks, which integrate topological and spatial features of a network. This study explored the potential to learn representations of the street network of the whole world. The node2vec model stands out as an efficient way thanks to its flexibility and simplicity, which does not require a pre-specified generative model of the graph.

This model implements biased random walks, which are used for a low-dimensional graph representation. The method is derived from the traditional random walk approach, which incorporates the parameters to play the role of direction-changing (breadth-first) and direction-preserving (depth-first) search, and thus aims to determine different features of the graph. Moreover, the synthetic map can be generated using the reconstructed adjacency matrices and the node sequence from the 0node model.

3. Methodology:

The main objective of this study is to understand the structure, connectivity and segmentation of street networks. These elements are useful for street network classification as well as for algorithmic approaches that generate artificial street networks. The studied embeddings are intended to identify both the spatial and geometric properties of street networks. To achieve this, street network data from cities around the world are collected and an undirected graph labelled G= (V, E) is constructed, where V and E correspond to intersections and streets, respectively. In addition, a node feature matrix X is constructed from the coordinates (latitude and longitude) of each node. Given the adjacency matrix A of G and the node feature vector X, new examples can be generated by learning a distribution over graphs G. The modelling task is divided into two components:

1) Understanding the distribution of coordinates over graphs and generating graph nodes with their coordinate pairs X.

2) Generation of an adjacency matrix A for the given nodes which indicates how the nodes are connected to each other.

This study implements graph representation learning on street networks using a different node and adjacency model. The node model employs the auto-regressive technique while, the adjacency model is based on the node2vec algorithm. The coordinate sequence and the adjacency matrix generated by these models will enable us to create synthetic graphs.

The decoder-only transformer of the node model is based on the fact that it needs to keep the input and output sequences consistent. This technique aligns with the goal of correctly modelling the basic structure of street networks, with the ultimate goal of smoothly integrating it into the next stages of our representation learning chain. By using different node and adjacency models and incorporating the transformer architecture, the aim is to construct diverse and accurate representations of street networks, covering both topological and spatial attributes. The synthetic street networks can be generated by sampling from the node model and then passing the edge list as input to the node2vec model.

3.1 Learnings on graph:

Graphs are simply the structures that mathematically describe how objects relate to each other. They are made up of nodes (also known as vertices) and edges that connect the nodes. Analyzing graphs is about drawing conclusions and finding patterns from the information they contain. This is an interdisciplinary field, with aspects drawn from fields such as machine learning, graph theory and linear algebra. There are techniques for learning from graph data, including:

- **Graph kernels:** These kernels give us a means of quantitatively measuring the similarities and contrasts between graphs. Such a measure can be used to categorise or group structurally similar graphs.
- **Graph neural networks:** The architecture of neural networks includes a graph-based structure that they process directly as it is presented to them. For example, they can be used for tasks such as classifying the nodes within a graph, categorising graphs themselves, and predicting the connections between nodes.
- **Graph representation learning:** Graph representation learning focuses primarily on creating meaningful annotations for both nodes and edges in a graph. These annotations are often low-dimensional vectors that capture node properties, relationship semantics, and the structural aspect well. Graph representation learning transforms continuous vector spaces with similarity metrics, unlike standard graphs which are discrete entities. Using this technique, nodes can be used for node classification, link prediction and node clustering in graphs by embedding their semantic information, where our research is mainly based this technique.

Most of the learning of graph representation in a street network scenario focuses on the extraction of granular and general knowledge needed for urban planning, traffic management and travel assistance systems. A graph representation of street networks is feasible, where intersections of street are represented by nodes and street segments are represented by edges. By using graph representation learning approaches generated by embedding intersections and street segments allow to identify significant topological and spatial attributes of the street network structure.

Street network will provide topology-specific properties such as connections between nodes and spatial relationships between street segments when representation learning concepts are applied to our graphs.

4. Application:

This module mainly describes in detail the data processing, implementation and the results structured after the implementation.

4.1 Input data and processing:

The data collection and processing involved in this research was inspired by (Neira and Murcio et al., 2022). The dataset in this research comes from OpenStreetMap (OSM), a freely accessible and comprehensive geographic database created and maintained by volunteers from all over the world. OpenStreetMap acts as a powerful repository of spatial data, containing detailed information about roads, buildings, historical landmarks and other geographic features. To ensure the completeness and suitability of the data for analysis, a well-defined data collection and pre-processing procedure is used. The initial stage of this methodological process is the compilation of a list of ISO-3166 countries and their associated regional codes. This list is integrated into the data collection process to encompass a range of geographical regions worldwide.

The comprehensive list of cities in every country is then generated using the OpenStreetMap's data query tool, the Overpass API. OpenStreetMap is queried to identify urban areas, and a list of cities is generated from which data on the urban road network can be obtained. The Overpass API allows users to collect city-scale data with minimal effort. This makes data collection and analysis easier and more efficient.

A city identification process is then used, which only considers cities with a population of over 1,000. This selection criterion ensures that only urban areas of significant population size are included in the analysis, making it easier to focus on regions with enough density to provide meaningful insights and results.

The centroid of each city is then determined using the geocoding functionality available in osmnx library (Boeing et al., 2017). The process of geocoding the positions of the city centers allows us to have a reference point from which we can extract the data which is essential. The data in the study is collected in a radius of 1 km by 1 km around selected city centres. Rather than gathering the entire city's street network, this method, as we understand it, makes sure that each city has an adequate amount of data to train the models without overloading them. The street network data is obtained in a tabular format with attributes such as type of highway, max speed, length and geometry, etc. Particularly relevant to our research is the geometry attribute, which represents each street as a line string. A line string is a sequence of straight-line segments, known as polylines, which together approximate the shape of the street. Each polyline segment simplifies the street’s path into a piecewise linear form, providing an accurate yet computationally manageable representation of street layouts. The line strings are processed to create a graphical representation of the city. This is then used in the 0node model.

With a total of 39,883 cities, the data is divided into training and test sets, with approximately 10% reserved for testing the model (i.e. 3909 cities), while the remaining cities were considered for training the model. This makes it easier to examine and evaluate the model. The preparation process and dataset provide a strong basis for urban analysis and modelling tasks for many scenarios. A full understanding of the complex spatial dynamics of the urban street network can be achieved by using OpenStreetMap data and following rigorous data collection and preparation techniques.

4.2 0node model:

The 0node model (Neira and Murcio, et al., 2022), is a task-specific graph representation learning method dedicated to the street network graph structure. The 0node method is different from the widely used general-purpose graph embedding methods, which are used for the analysis of the global urban networks, local traffic dynamics, and urban features. Here is how the 0node model works:

- **Input representation:** The 0node model requires street network data as an input and transforms it into a sequence of nodes or segments.
- **Transformer decoder architecture:** The 0node model uses the decoder part of the transformer only, in such a way that the input and output sequences are the same. The decoder processes the incoming sequences on an autoregressive basis; thus, the output sequence of each node is conditioned on the preceding nodes in the sequence.
- **Sequential context encoding:** The positional embedding in the transformer facilitates the model's understanding of the spatial relationships between the nodes in the street network. By concentrating on the preceding nodes in the sequence, the decoder gains the ability to understand the contextual information for each node's sequence position.

The model makes the representation learning procedure specific to the features of street networks, which is essential for proper downstream tasks such as urban planning and transportation management.

4.2.1 Details on execution:

The 0node model tries to capture a distribution over a sequence of nodes. After processing the data and creating graphs for each city, we create a node feature matrix that contains the x and y coordinates for each node as the features. We center each street network at (0,0) and normalize both x and y coordinates such that their bounding box diagonal is equal to 1. The next step is to apply 8-bit quantization to the centered and normalised coordinates. According to our understanding, 8-bit quantization is chosen to transform continuous coordinate values into discrete, categorical distributions that are influenced by its successful application in related fields. This approach standardizes coordinate values to a uniform scale from 0 to 255, simplifying data input to the transformer model while aiding data regularisation. Despite its simplicity, this level of quantization effectively preserves the essential spatial characteristics of the network, as evidenced by nodes falling into distinct bins within a spatial extent of 1 km. Furthermore, this strategy mirrors techniques used in 3D network modelling and continuous signal discretization, as discussed in (Nash et al. 2020) and (Van Oord et al. 2016), respectively.

The subsequent step is a structured sorting process. Initially, the nodes are arranged sequentially based on their y-coordinate values to establish a primary order. In instances where multiple nodes share identical y-values, a secondary sorting criterion is applied using the x-coordinates. This two-tiered sorting mechanism ensures a systematic arrangement of nodes from the lowest to the highest values, facilitating the subsequent computational steps. Following the sorting procedure, the ordered coordinates are transformed into a flattened sequence. This sequence, referred to as C^{seq}, represents a concatenation of coordinate pairs (x_i, y_i), which are streamlined to form a single-dimensional array. This transformation is of significant importance in the context of modelling spatial relationships between nodes, as it simplifies the complex multi-dimensional data into a format that is suitable for sequential processing. The flattened array, C^{seq}, is then created for each city in the dataset, which is then combined into a master C^{seq} file. This master file serves as the foundation for additional research and analysis, which is divided into discrete subsets for training, testing, and validation purposes.

The decomposition of this coordinate sequence forms the basis for constructing a joint distribution over the elements. Each element's distribution is contextually dependent on its predecessors, forming a series of conditional distributions. This auto-regressive framework is pivotal for employing a transformer architecture, as proposed by (Vaswani et al., 2017), which efficiently learns the spatial distribution of node coordinates. The selection of this methodology is consistent with the objective of capturing the inherent spatial dependencies within the dataset, thereby enabling a more nuanced understanding and representation of the underlying geographic structures.

The central element of 0node model is training the transformer through decoder-only approach. In this transformer structure, positional embedding is employed to encode the spatial information of the quantized coordinates in the sequences. This strategy will use the sequential nature of street network data and consistency between input (C^{seq}) and output co-ordinate sequence ($C^{seq'}$). The transformer model, adapted from (Karpathy et al., 2023), was customized to accommodate our specific dataset requirements. The training was conducted on the master C^{seq}, which includes data from 35,974 cities globally. For the computational resources, we utilized an NVIDIA A100 GPU, which was accessed through a cloud service. The model underwent extensive training over 20,000 iterations, spanning approximately 40 hours. This rigorous training process achieved a final loss metric of approximately 0.4.

Figure 1: 0node model

4.3 node2vec model:

node2vec, is a semi-supervised technique for scalable feature learning in networks. The method concentrates on maximizing the likelihood of preserving network neighborhoods of nodes in a d-dimensional feature space. A second-order random walk is applied to generate sample nodes' network neighborhoods. node2vec can learn the representation of node embedding based on their network roles and the areas they belong to. This is achieved by creating biased random walks that indicate the diversity of nodes' neighborhoods.

There are two sampling strategies for creating neighborhood sets of a source node:

Breadth-first Sampling (BFS): The neighborhood is limited to nodes that are direct neighbors to the source.

Depth-first Sampling (DFS): The neighborhood includes nodes sampled at increasing distances from the source node.

BFS and DFS plays an important role in creating embeddings that reflects homophily and structural equivalence (connected nodes should be embedded closely together). BFS identifies nearby nodes and provides a detailed view of their surroundings. Additionally, in BFS, nodes in sampled neighbourhoods frequently repeat. This lowers variation in describing the distribution of 1-hop nodes based on the source node. However, only a small section of the graph is investigated for each k. DFS, on the other hand, can explore more of the network by moving away from the source node (with a fixed sample size of k). DFS provides a more realistic representation of the area, which is crucial for identifying homophilic populations.

The working principle of node2vec involves two key steps: biased random walks and learning embeddings through Skip-gram.

Random walks: for a given source node u, a random walk of fixed length l is simulated. Let c_i denote the i^{th} node in the walk, starting with $c_0 = u$. Nodes c_i are generated by the following distribution:

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

where π_{vx} is the unnormalized transition probability between nodes v and x, and Z is the normalizing constant.

The second-order random walk is defined with two parameters p and q. The unnormalized transition probability is set to $\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$, where:

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

and d_{tx} denotes the shortest path distance between nodes t and x.

Return parameter, p: determines the possibility of immediately revisiting a node throughout the walk.

In-out parameter, q: differentiates the search between “inward” and “outward” nodes. From a street network perspective, biased random walks involve solving the street network graph by moving from one node to another that efficiently reveals the graph neighbourhood exploration through the local and global graph structure information by considering factors such as road connectivity, proximity, and popularity of routes. These random explorations express the local structure of the network and allow the algorithm to travel through different parts of the graph.

Algorithm 1 The *node2vec* algorithm.

```
LearnFeatures (Graph G = (V, E, W), Dimensions d, Walks per node r, Walk length l,
Context size k, Return p, In-out q)
    π = PreprocessModifiedWeights(G, p, q)
    G' = (V, E, π)
    Initialize walks to Empty
    for iter = 1 to r do
        for all nodes u ∈ V do
            walk = node2vecWalk(G', u, l)
            Append walk to walks
    f = StochasticGradientDescent(k, d, walks)
    return f

node2vecWalk (Graph G' = (V, E, π), Start node u, Length l)
    Initialize walk to [u]
    for walk_iter = 1 to l do
        curr = walk[−1]
        V_curr = GetNeighbors(curr, G')
        s = AliasSample(V_curr, π)
        Append s to walk
    return walk
```

Following random walks, node2vec uses strategies such as Skip-gram to discover an embedding space that captures the structural similarities between nodes. Skip-gram predicts the probability of finding neighbouring nodes during the random walk of the current node. The optimization is implemented to maximize the log-probability of observing a network neighbourhood $N_S(u)$ for a node u conditioned on its feature representation, given by f:

$$\max_{f} \sum_{u \in V} \log Pr(N_S(u) | f(u))$$

Through such a prediction task, node2vec learns the embeddings that can capture the topology of the network, enabling downstream tasks such as node classification, link prediction, and community detection.

4.3.1 Details on execution:

This research evaluates the feature representations obtained through node2vec on standard supervised learning tasks: multi-label classification for nodes and link prediction for edges. The execution of the node2vec model contained several steps and required the setting of various parameters. This model took the graph data in the form of an edge list (the list that contains the pairs of nodes connected) as input. With the dimensions parameter set to 128, therefore for each node, it has a 128-dimensional vector in the embedding. This model also learns representation by performing random walks on the graph. To adapt the node2vec model in our research, we employed the length of walks and number of walks parameters, setting $l = 50$, and $r = 20$. This implied that for each source node, the model performed 20 random walks, each with a length equal to 50. We set the window size parameter, which defines the context size for optimization, to 5 and determined how many nodes on either side of a given node in a walk sequence were considered its context during the learning process. We trained the model with 100 epochs in the Stochastics Gradient Descent (SGD) optimization process for learning the best embeddings. For the two most important hyperparameters in this model; the return p and in-out q hyperparameters were set to 20. These factors determine the possibility of quickly revisiting a node in a walk and exploring outward nodes, influencing the diversity of random walks and subsequent embeddings. The weighted and directed parameters determined whether the graph was weighted or directed. These were determined based on the characteristics of our data set.

The node2vec model was executed after setting all the parameters. This process involved reading the graph, calculating transition probabilities, simulating random walks, and finally, training the Skip-gram model to learn the embeddings.

In this case, the Skip-gram model aimed to learn contextually similar embedding for nodes by optimizing a neighborhood that had a likelihood objective. The Skip-gram objective is based on the distributional hypothesis which states that connected nodes tend to have similar values.

The output embeddings, capturing the topological information of the nodes in a low-dimensional space, served as a comprehensive feature set that could be used in various downstream machine-learning tasks, enabling us to leverage the rich structural information of our graphs for further analysis.

After having the node embedding, which represents the connection between nodes, we converted it to a reconstructed adjacency matrix. Here we set a threshold equal to 0.876 to decide whether two nodes are connecting or not. This value was chosen by balancing a meaningful level of connectivity in the synthetic map and its structural integrity. The purpose of setting the threshold at 0.876 is to capture the relationships between nodes while minimizing false connections and improving the overall interpretability of our generated synthetic street networks.

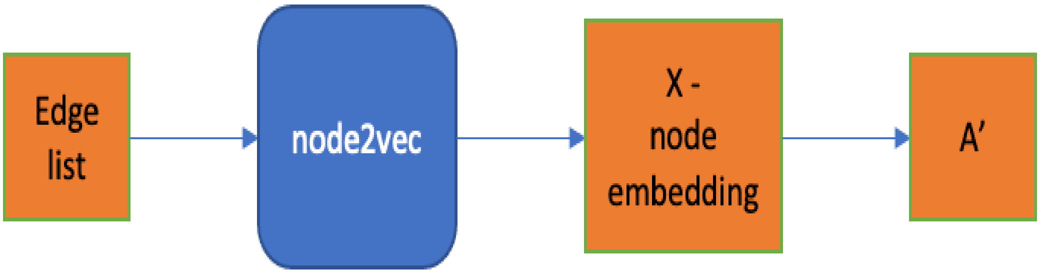
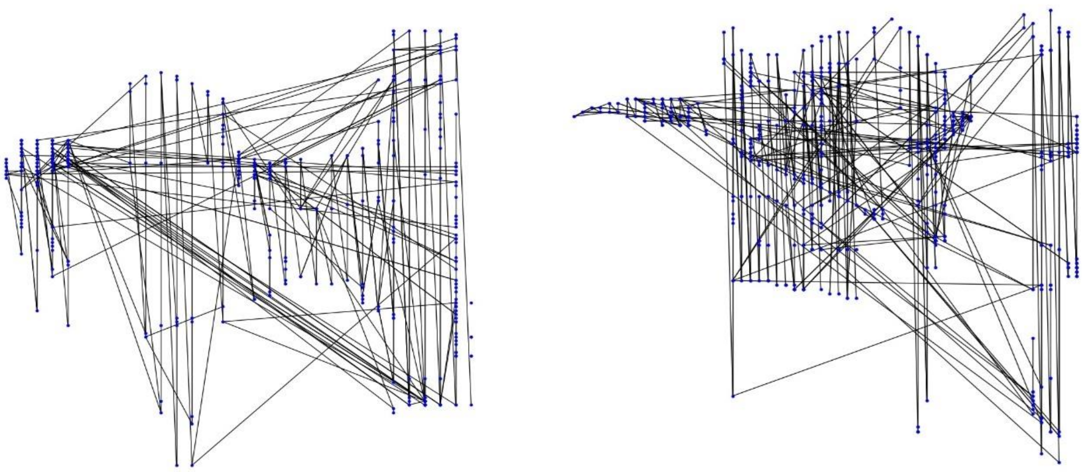


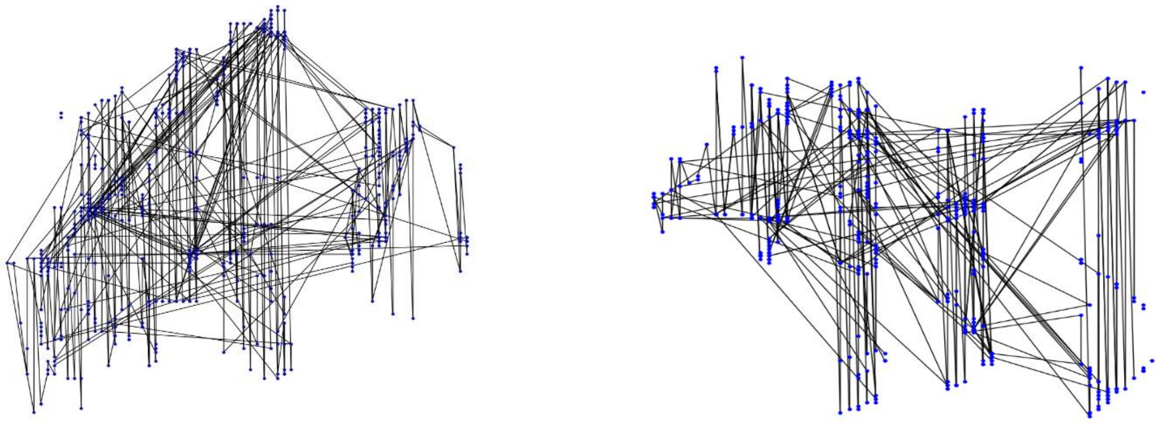
Figure 2: node2vec model

4.4 Results and feature exploration:

We evaluated the performance of our model by analyzing its ability to generate artificial road networks. We reconstruct synthetic street networks to assess the quality of the network reconstruction. The street networks in our test dataset, which represents 10% of the total data, are compared with the reconstructed street networks. An example of a synthetic street network from German cities.



Dortmund **Dresden**



Heidelberg Trier

Figure 3: Synthetic street networks of Germany

Figure 3 shows the generated synthetic street networks of (Dortmund, Dresden, Heidelberg, Trier) Germany within 1 square km from the city center. These maps exhibits some characteristics in terms of capturing important urban aspects like road layout and spatial connection. However, some refinements are still needed to improve the performance of the model. In this research, it was observed that the model predicts a large number of streets as parallel lines. The specific causes of this are difficult to determine, as they depend on the details of the model's internal learning processes, as well as the properties of the data it encounters. To enable more accurate and thorough urban planning and decision making, future work needs to focus on the model's predictive power, scalability and computational efficiency.

Topological features

This section outlines a detailed comparative analysis of graph summaries taken from real street networks and those taken from a model. Through an extensive sampling process that includes 412 cities from both the test dataset and the model, the research examines basic metrics such as average edges per node and average circulation, which is a measure of network efficiency. Although these measures are inherently simplistic in their attempt to capture the complexity of street networks, the model's output shows distributions that are similar to those of the real data, suggesting a correspondence in their underlying structures.

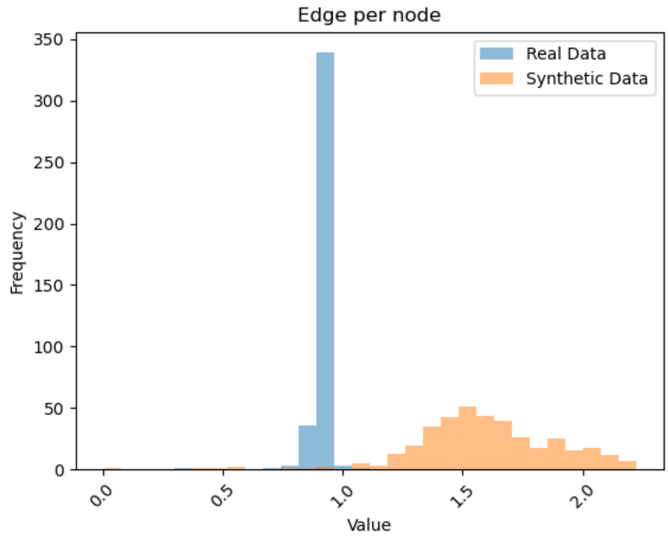


Figure 4: Average edges per node

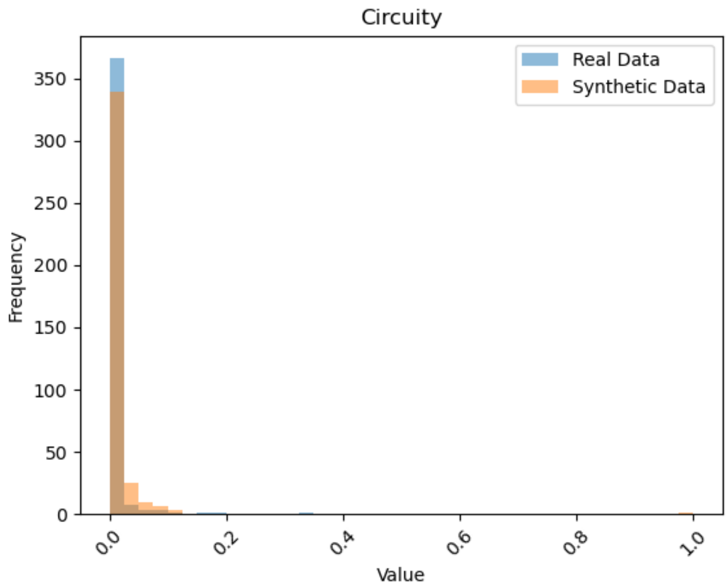


Figure 5: Average circuity

The distribution of topological features in generated and real street network samples are shown in the above figures. The average number of edges per node and the average circulation, which is determined by the ratio of the network distance to the Euclidean distance between two nodes. After the clear analysis we can confirm that the model replicates the components of real-world urban systems and explain the intricate relationship between the topological features and graph-based learning in street network models.

Geometric features

The analysis focuses on a comparison of the geometric features embedded in the created street network. This objective is realised through the faces of the planar graph, which correspond to 412 city samples. After extracting the faces from each of the graphs, belonging to both real and synthetic street networks, we proceeded to compute the basic geometric metrics to describe their spatial properties. In particular, two important metrics are used to evaluate the characteristics of the street network components: the average compactness, which is determined by dividing each block's perimeter length by area, and the average form factor, which is calculated by dividing a block's area by the area of its bounding rectangle as a reference. These visualizations serve as substitutions for the real spatial arrangement of blocks and streets.

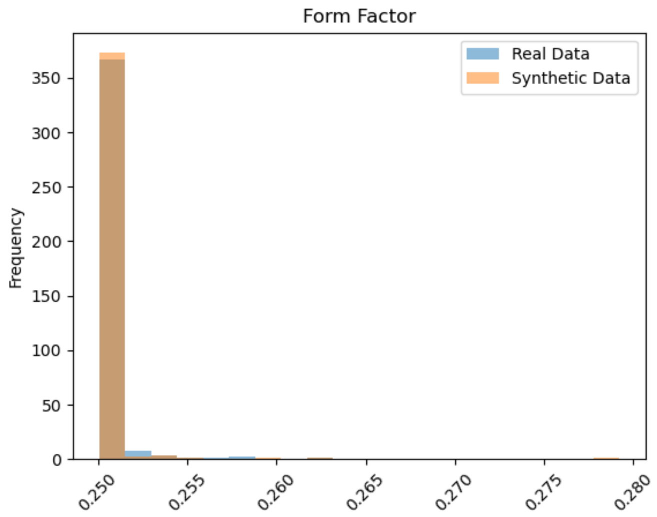


Figure 6: Average form factor

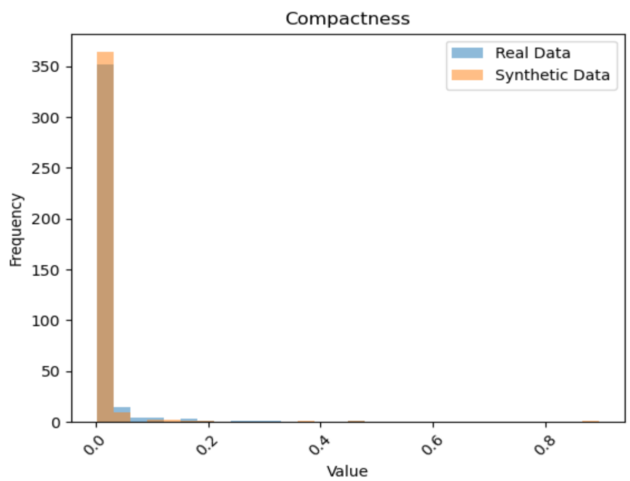


Figure 7: Average compactness

This study aimed to prioritize computational time and cost, while effectively managing computational resources by selecting a subset of the test dataset. This is because the creation of synthetic data requires a lot of computational power just by using the created data and using cloud resources, while staying within the project budget. The distributions of geometric features in both real and synthetic data can be determined by comparing them which represents the accuracy of the model. This analysis highlights the complex interactions between geometric and topological properties in learning road network graph representations. In addition, it highlights how these properties enable the model to accurately represent the complex spatial texture of real-world environments.

Finally, the results highlight the importance of topological and geographical aspects in understanding street networks. The synthesis of these features through advanced learning techniques provides a powerful framework for comprehensively characterizing urban environments.

5. Outlook:

This study followed the method of transforming line strings into street network graphs, the key aspect being that they are now expressed as graphs rather than line shapes. The low resolution of the concrete and the curving nature of the streets impose restrictions on such a simplification. The accuracy of the simulation may be indirectly impacted by this simplification if the street curvatures are inaccurate.

The way forward is to explore and discover the range of modified graph construction techniques that offer the possibility of a better representation of the street network. There is one avenue that looks very promising, and that is to develop systems and methods that take into account the curvature of streets and also intersections accurately. Therefore, the aim of this study is to reproduce the actual characteristics of the urban street layout using our graph representation; this will lead to an improvement in the accuracy and reliability of the predictions.

It is also important to find solutions to the computational problems that our research presents. We understand the inevitability of resource constraints, such as lack of access to advanced GPUs for training more complex models. The computational cost of improvement and the commercial cost of model training are significant barriers to achieving the desired results. On the other hand, access to high-end computing facilities with advanced GPUs could be a solution to overcome the limitations and train more appropriate models that can better capture the complexity of street networks.

Thus, research on graph learning on street networks has brought certain aspects of modelling civic infrastructure into the picture. In this sense, our approach has been successful in providing many insights, but there is still a need to fine-tune the graph construction and work out some computational challenges. By increasing the use of alternative architectural techniques and taking full advantage of today's advanced computing tools, our models will become more accurate and relevant, allowing for more thoughtful decisions in urban planning and infrastructure design.

References:

Boeing, G., 2017. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. Computers, environment and urban systems, Volume 65, p. 126–139.

Contributors, O., 2017. Planet Dump Retrieved from https://planet. osm. org. URL: https://planet. openstreetmap. org.

Grover, A. & Leskovec, J., 2016. node2vec: Scalable feature learning for networks. s.l., s.n., p. 855–864.

Hartmann, S., Weinmann, M., Wessel, R. & Klein, R., 2017. Streetgan: Towards road network synthesis with generative adversarial networks.

Karpathy, A., n.d. Nanogpt, 2023. URL https://github. com/karpathy/nanoGPT.

Kempinska, K. & Murcio, R., 2019. Modelling urban networks using Variational Autoencoders. Applied Network Science, Volume 4, p. 1–11.

Louf, R. & Barthelemy, M., 2014. A typology of street patterns. Journal of the royal society Interface, Volume 11, p. 20140924.

Marshall, S. et al., 2018. Street network studies: from networks to models and their representations. Networks and Spatial Economics, Volume 18, p. 735–749.

Murcio, R., Masucci, A. P., Arcaute, E. & Batty, M., 2015. Multifractal to monofractal evolution of the London street network. Physical Review E, Volume 92, p. 062130.

Nash, C., Ganin, Y., Eslami, S. A. & Battaglia, P., 2020. Polygen: An autoregressive generative model of 3d meshes. s.l., s.n., p. 7220–7229.

Neira, M. & Murcio, R., 2022. Graph representation learning for street networks. arXiv preprint arXiv:2211.04984.

Strano, E. et al., 2012. Elementary processes governing the evolution of road networks. Scientific reports, Volume 2, p. 296.

Van Den Oord, A., Kalchbrenner, N. & Kavukcuoglu, K., 2016. Pixel recurrent neural networks. s.l., s.n., p. 1747–1756.

Vaswani, A. et al., 2017. Attention is all you need. Advances in neural information processing systems, Volume 30.