TØIRoads: A Road Data Model Generation Tool

Grunde Haraldsson Wesenberg
Department of Informatics, University of Bergen, Norway
Institute of Transport Economics, Oslo, Norway

Ana Ozaki
Department of Informatics, University of Oslo, Norway
Department of Informatics, University of Bergen, Norway

Abstract

We describe road data models which can represent high level features of a road network such as population, points of interest, and road length/-cost and capacity, while abstracting from time and geographic location. Such abstraction allows for a simplified traffic usage and congestion analysis that focus on the high level features. We provide theoretical results regarding mass conservation and sufficient conditions for avoiding congestion within the model. We describe a road data model generation tool, which we call "TØI Roads". We also describe several parameters that can be specified by a TØI Roads user to create graph data that can serve as input for training graph neural networks (or another learning approach that receives graph data as input) for predicting congestion within the model. The road data model generation tool allows, for instance, the study of the effects of population growth and how changes in road capacity can mitigate traffic congestion.

2012 ACM Subject Classification General and reference → Evaluation

Keywords and phrases Road Data, Transportation, Graph Neural Networks, Synthetic Dataset Generation

Digital Object Identifier 10.4230/TGDK.2.2.6

Category Resource Paper

Supplementary Material *Software (Source Code)*: `https://github.com/gruwesen/TOIROADS` [19]
archived at `swh:1:dir:a86388944844fcc00f4cad67e1ec75a998f36eae`

Funding *Grunde Haraldsson Wesenberg*: The author is supported by the Norwegian Research Council, project 322480.
Ana Ozaki: The author is supported by the Norwegian Research Council, projects 322480, 316022. This work was partly supported by the Research Council of Norway through its Centre of Excellence Integreat - The Norwegian Centre for knowledge-driven machine learning, project number 332645.

Received 2024-07-02 **Accepted** 2024-11-08 **Published** 2024-12-18

Editors Aidan Hogan, Ian Horrocks, Andreas Hotho, Lalana Kagal, and Uli Sattler

Special Issue Resources for Graph Data and Knowledge

1 Introduction

Modelling and predicting traffic is fundamental for the administration of a city and its surroundings as it directly impacts the economy and everyday lives of the population [11, 2, 9, 1]. While works on traffic prediction tend to focus on short-term prediction (e.g., 12 steps of 5 minutes in the future) [14, 18, 5], a high-level view of the main properties of a road network is crucial for long-term city planning. For instance, one may want to plan for the increase of road capacity by enlarging road segments or to plan for the development of new links in the road network if a systematic congestion scenario is foreseen due to an increase of the population or significant changes in traffic flow resulting from the construction of new points of interest in the city.

The solutions to the high-level setting are classically explored in the context of agent-based traffic simulators [16]. However, agent-based traffic simulators, such as MATSIM [10], are co-evolutionary tools which inherently cannot be parallelized. The complete configuration of features and parameters used in the real world traffic simulators include the specification of a number of agent-based traffic routines. This complexity makes their format hard to use in machine learning approaches, which are parallelizable. Machine learning approaches, in particular those based on graph neural networks, are useful for predicting what would happen in these scenarios [12, 20, 6], however, they need datasets with the relevant features.

In this work we present a road data model generation tool, which we call "TØIRoads", that can create datasets synthetically, based on user defined parameters. We describe road data models which can represent high level features of a road network such as population, points of interest, and road length/cost and capacity. Such abstraction allows for a simplified traffic usage and congestion analysis that focus on the high level features. We provide theoretical results regarding mass conservation and sufficient conditions for avoiding congestion within the model.

A main benefit of generating road data synthetically is that the data is independent of any particular city structure. Datasets of varying sizes can be created, simulating the fact that road networks also vary in size. Moreover our road data generator has the interesting feature that the user can give a particular structure (which could realistically represent the information from a real world road network) and use our road data model generation tool to create variants ("mutations") of this network, e.g., with increasing population, which could indicate possible future congestion issues if the population grows without an increase of capacity in the road network.

In Section 3, we describe our road data models and establish some theoretical properties related to these data models. In Section 4, we describe several parameters that can be specified by a TØIRoads user to create graph data that can serve as input for training graph neural networks (or another learning approach that receives graph data as input) for predicting congestion within the model. In Section 5 we present some statistics associated with the datasets generated by TØIRoads. Finally, we conclude in Section 6.

2 Related Work

Traffic prediction is a broad research field that has been heavily studied in the literature [3, 11, 13]. Most works focus on short term or even real time traffic prediction, with the goal of supporting drivers to find an optimal path. Indeed, as many cars and drivers today are sending their information to the internet, some researchers have been very successful in predicting traffic based on real time driver resolution data [7].

Jiang and Luo (2022) describe some datasets used for traffic prediction with graph neural networks as well as what variants of graph neural networks are used for the prediction tasks [11]. They categorize studies by scale and domain, and include passenger flow for bus and metro systems, as well as various studies focusing on road traffic flow, speed, congestion and other variables. What is common among the road traffic datasets is the usage of counting stations in city road systems. Counting stations can have different time resolution as well as various measurements in addition to simple counting of vehicles. Olug et al. (2024) [17] show for a dataset like this that adding features, such as population density and districts, to the counting station data points can greatly improve predictions. These datasets related to various modes of transportation are based on available data and as such are quite relevant to traffic prediction research.

Bui et al writes about using spatial temporal graph neural networks for traffic forecasting [3], and points to how to construct the adjacency matrix of a graph based on traffic counting station nodes is an open problem, in terms of effectively capturing information streams. For instance, in the urban vehicle detection system dataset [4] that they use, there is no single way to make the adjacency matrix, as there are many road links between each counting station. The TØIRoads dataset generator can take as input a graph structure based on real world data and make datasets with variants of the original data, and so, it might be used as a tool for studying this graph construction problem.

Here we consider traffic from the point of view of road administrators, also called road planners, who have the goal of predicting the need e.g., of improving road capacity in certain road segments to avoid congestion in the city. Our model is made to work as an aggregate of traffic data, representing high level features of a road network, such as population, points of interest, and road length/cost and capacity, while representing the graph structure in detail, at a road segment level. Our model is not restricted to a particular graph structure and it can generate graph structures synthetically. There are various works in the literature on generating graph data synthetically [8, 15], however, here we use features and structural constraints that are meaningful for road traffic prediction. We prove theoretical results related to traffic conservation and congestion for our road data models.

3 Modelling Road Traffic

In this section we describe our road data models. The model is an abstraction of the road traffic conditions found in real world scenarios, useful to estimate road usage and congestion. The road network is represented with a graph, where nodes are road segments and edges connect road segments which would be connected in the road network. High population contributes to increase the traffic of a road segment, while high cost contributes to decreasing the traffic. Road capacity is a feature that combines with the measure of traffic to determine if a road segment is congested or not. See Section 4.3 for a detailed example containing the notions presented here.

3.1 Formal Definition

We formally define a road traffic model based on the components related to road usage and congestion described above. We call this model RoadGNN.

▶ **Definition 1** (RoadGNN). *A* RoadGNN road data model *is a labelled graph $G = (V, E, L)$, where V is a finite set of nodes representing road segments, E is a (finite) set of edges representing connections between road segments, and L is a labelling function mapping nodes to their respective attributes. Each node $v \in V$ represents a road segment and v is associated via the labelling function L with the following attributes:*

- *road weight $w_v \in \mathbb{R}$ with $w_v > 0$,*
- *capacity $c_v \in \mathbb{R}$ with $c_v > 0$,*
- *population $p_{v,t} \in \mathbb{N}$ from node v to each node $t \in V$,*

where w_v is a weight value associated with node v, depending on its length/cost; c_v quantifies the road capacity associated with node v, and $p_{v,t}$ represents the flow with origin in v and destination in t. In symbols, $L(v) = (w_v, c_v, p_{v,t_1}, \ldots, p_{v,t_n})$ with $t_1, \ldots, t_n \in V$.

In the following we consider a fixed but arbitrary labelled graph G. We omit G from the notation to simplify it since there is no risk of confusion. A *path* from $s \in V$ to $t \in V$ is a sequence of nodes $v_1, \ldots, v_k \in V$ with $s = v_1$, $t = v_k$, and $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$. We define a total order on paths where path p_1 is shorter than p_2, in symbols $p_1 < p_2$, if the sum of the weights $w_{v_1}, \ldots, w_{v_k}$ of the nodes $v_1, \ldots, v_k$ in p_1 is smaller than that sum in p_2.

▶ Remark 2. All labelled graphs we consider in this work are *strongly connected*, that is, they have the property that for every pair of nodes $(s, t) \in V \times V$ there is a path from s to t.

For each pair of nodes (s,t), define $S_{s,t}$ as the set of shortest paths between s and t. Also, we denote by $S^v_{s,t}$ the subset of paths in $S_{s,t}$ where $v \in V$ occurs. Given a finite set S, we write $|S|$ for the number of elements in S. By definition, sets of shortest paths in the graph are finite. Given a node $v \in V$, we write $\mathsf{in\_pop}(v)$ as a shorthand for $\sum_{s\in V} p_{s,v}$ and $\mathsf{out\_pop}(v)$ as a shorthand for $\sum_{t\in V} p_{v,t}$. Let $\mathsf{in}(v)$ and $\mathsf{out}(v)$ denote the sets of incoming and out going nodes from node v.

We now describe how we estimate the values associated with road usage and congestion. We assume that agents aim at trips that minimize road usage, by sticking with shortest paths when calculating a route.

▶ **Definition 3** (Road Usage). *The road usage of node $v \in V$, denoted U_v, is defined as follows:*

$$\sum_{(s,t)\in V^2} p_{s,t} \cdot \frac{|S^v_{s,t}|}{|S_{s,t}|}.$$

In real traffic, a road is not congested when the traffic is below capacity, and congested when the traffic is above capacity. We define our congestion coefficient as the quotient between road usage and capacity, and call the node congested if this is 1 or more.

▶ **Definition 4** (Congestion). *Congestion is the ratio between road usage and capacity, in symbols,*

$$C_v := \frac{U_v}{c_v}.$$

We say that a node v is congested *if $C_v \geq 1$.*

▶ Remark 5. By definition of U_v, we have that $U_v \leq \mathsf{in\_pop}(v) + \mathsf{out\_pop}(v)$.

We now establish theoretical bounds regarding traffic conservation and congestion.

▶ **Theorem 6** (Conservation). *For all $v \in V$,*

1. $U_v \leq \sum_{u\in \mathsf{in}(v)}(U_u - \mathsf{in\_pop}(u)) + \mathsf{out\_pop}(v)$
2. $U_v \leq \sum_{u\in \mathsf{out}(v)}(U_u - \mathsf{out\_pop}(u)) + \mathsf{in\_pop}(v)$.

Proof. We start proving Point 1. Let v be a node in V. By definition of U_v, we can write U_v, as follows:

$$\sum_{(s,t)\in (V\setminus\{v\})\times V} p_{s,t} \cdot \frac{|S^v_{s,t}|}{|S_{s,t}|} + \sum_{(v,t)\in \{v\}\times V} p_{v,t} \cdot \frac{|S^v_{v,t}|}{|S_{v,t}|}.$$

By definition, $S^v_{v,t}$ is the set of shortest paths from s to v that passes through v, which coincides with the set of shortest paths from s to v, denoted $S_{v,t}$. In other words, $S^v_{v,t} = S_{v,t}$. So U_v is

$$\sum_{(s,t)\in (V\setminus\{v\})\times V} p_{s,t} \cdot \frac{|S^v_{s,t}|}{|S_{s,t}|} + \mathsf{out\_pop}(v).$$

It remains to show that

$$\sum_{(s,t)\in (V\setminus\{v\})\times V} p_{s,t} \cdot \frac{|S^v_{s,t}|}{|S_{s,t}|} \leq \sum_{u\in \mathsf{in}(v)} (U_u - \mathsf{in\_pop}(u)).$$

Given a node v and pair of nodes (s,t), if a path p is in $S^v_{s,t}$ and $s \neq v$ then p is in $S^u_{s,t}$ for some $u \in \mathsf{in}(v)$. Then,

$$\sum_{(s,t)\in (V\setminus\{v\})\times V} p_{s,t} \cdot \frac{|S^v_{s,t}|}{|S_{s,t}|} \leq \sum_{u\in \mathsf{in}(v)} \sum_{(s,t)\in V^2} p_{s,t} \cdot \frac{|S^u_{s,t}|}{|S_{s,t}|}.$$

In other words,

$$\sum_{(s,t)\in(V\setminus\{v\})\times V} p_{s,t}\cdot\frac{|S^v_{s,t}|}{|S_{s,t}|}\leq\sum_{u\in\mathsf{in}(v)} U_u.$$

Moreover, we have that $u \neq t$ since by assumption there is a path in $S^v_{s,t}$ (in other words, v belongs to a shortest path between s and t, which would not be the case if $u = t$ since $u \in \mathsf{in}(v)$). So we can subtract the population associated with those paths:

$$\sum_{(s,t)\in(V\setminus\{v\})\times V} p_{s,t}\cdot\frac{|S^v_{s,t}|}{|S_{s,t}|}\leq\sum_{u\in\mathsf{in}(v)} (U_u-\sum_{s\in V}p_{s,u}).$$

This means that

$$\sum_{(s,t)\in(V\setminus\{v\})\times V} p_{s,t}\cdot\frac{|S^v_{s,t}|}{|S_{s,t}|}\leq\sum_{u\in\mathsf{in}(v)} (U_u-\mathsf{in\_pop}(u))$$

as required. The proof of Point 2 is analogous. ◀

We now provide sufficient (but not necessary) conditions for preventing node congestion.

▶ **Theorem 7** (Outgoing Congestion)**.** *For all $v \in V$, if*

1. $c_v \geq \sum_{u\in\mathsf{out}(v)} c_u + \mathsf{in\_pop}(v)$, *and*
2. $\forall u \in \mathsf{out}(v)$, $C_u < 1$,

then v is not congested.

Proof. We can rewrite Assumption 1 as

$$c_v \geq \Delta_v + \sum_{u\in\mathsf{out}(v)} U_u + \mathsf{in\_pop}(v) \tag{1}$$

where

$$\Delta_v := \sum_{u\in\mathsf{out}(v)} (c_u - U_u).$$

By Point 2 of Theorem 6,

$$U_v \leq \sum_{u\in\mathsf{out}(v)} (U_u - \mathsf{out\_pop}(u)) + \mathsf{in\_pop}(v).$$

Adding $\Delta_v + \sum_{u\in\mathsf{out}(v)} \mathsf{out\_pop}(u)$ on both sides,

$$U_v + \Delta_v + \sum_{u\in\mathsf{out}(v)} \mathsf{out\_pop}(u) \leq \Delta_v + \sum_{u\in\mathsf{out}(v)} U_u + \mathsf{in\_pop}(v).$$

By Eq. 1,

$$U_v + \Delta_v + \sum_{u\in\mathsf{out}(v)} \mathsf{out\_pop}(u) \leq c_v.$$

By Assumption 2, $\forall u \in \mathsf{out}(v)$, $c_u > U_u$. Thus, $\Delta_v > 0$. Since $\sum_{u\in\mathsf{out}(v)} \mathsf{out\_pop}(u) \geq 0$, we have that $c_v > U_v$. In other words, v is not congested. ◀

We also have an analogous theorem for the incoming congestion, which can be proved in a similar way as for outgoing congestion.

▶ **Theorem 8** (Incoming Congestion)**.** *For all $v \in V$, if*

1. $c_v \geq \sum_{u\in\mathsf{in}(v)} c_u + \mathsf{out\_pop}(v)$, *and*
2. $\forall u \in \mathsf{in}(v)$, $C_u < 1$,

then v is not congested.

3.2 A Practical Special Case

In the model just described each node v has a feature $p_{v,t}$ associated with each other node t in the graph, which quantifies the population going from v to t. This means that the number of features in a node grows depending on the size of the graph (in particular, in the number of nodes), which is not ideal in practice. So here we consider a simplified model, called S-RoadGNN, which associates a fixed number of features (4 features, namely, road weight, capacity, population, and points of interest) to each node. The idea is that people live along roads (population), and they travel to points of interest (POI) that are along the roads. The traffic associated with each node is given by the amount of transport demand density and it depends on its close and far neighbours. Although this simplification takes away some of the expressivity of our road data models (see Remark 12), it is flexible enough to create a range of road data models which are useful to study the effects of population growth and how changes in road capacity can mitigate traffic congestion.

▶ **Definition 9** (S-RoadGNN). *A* S-RoadGNN road data model *is a labelled graph $G = (V, E, L)$, where V is a finite set of nodes representing road segments, E is a (finite) set of edges representing connections between road segments, and L is a labelling function mapping nodes to their respective attributes. Each node $v \in V$ has the following attributes:*

- *road weight $w_v \in \mathbb{R}$ with $w_v > 0$,*
- *capacity $c_v \in \mathbb{R}$ with $c_v > 0$,*
- *population $p_v \in \mathbb{N}$,*
- *points of interest $i_v \in \mathbb{N}$,*

where w_v is a weight value associated with node v, representing a road segment, depending on its cost/length; c_v quantifies the road capacity associated with node v; and finally, p_v and i_v are associated with v depending on the population and on the points of interest in the vicinity of the road segment represented by v. In symbols, $L(v) = (w_v, c_v, p_v, i_v)$.

▶ **Definition 10** (Road Usage). *The road usage of node $v \in V$, denoted U_v, is defined as follows:*

$$\sum_{(s,t)\in V^2} p_s \cdot i_t \cdot \frac{|S^v_{s,t}|}{|S_{s,t}|}.$$

▶ **Definition 11** (Congestion). *Congestion is as in Definition 4: the ratio between road usage and capacity. A node v is* congested *if $C_v \geq 1$.*

▶ Remark 12. S-RoadGNN can be seen as a special case of RoadGNN where $p_{s,t} = p_s \cdot i_t$. In RoadGNN, we can have models where e.g., there is traffic from a node s to a node t, but no traffic from another node u that has some population to t. This is not possible in S-RoadGNN because in this model whenever a node u has some population and another node t has some points of interest, there is some traffic from u to t, namely $p_u \cdot i_t$.

We formalize this remark with the following theorem.

▶ **Theorem 13.** *Let G be an S-RoadGNN road data model and let G' be the RoadGNN road data model that is defined in the same way as G, except that $p_{s,t} := p_s \cdot i_t$ for all $s, t \in V$. For all $v \in V$, $U_v = U'_v$, where U_v and U'_v correspond to the road usage of v in G and G', respectively.*

It follows from Theorem 13 that the conservation and congestion bounds established for RoadGNN also hold for S-RoadGNN. We are now ready to describe TØIRoads, which is a tool for generating S-RoadGNN road data models.

4 TØIRoads: Road Data Model Generation

Here we present our road data model generator, named TØIRoads, for simulating traffic conditions in a simplified road network. The road data models generated by TØIRoads are instances of the model described in Section 3.2. We first describe the parameters that the user can give and then the procedure for generating road data models according to the parameters given by the user.

4.1 Parameters for Road Data Model Generation

The user can give as input the *number of road data models* to be created, together with the *minimum and maximum number of nodes* allowed in each road data model. The user can also influence the *graph density*, which is the ratio between the number of edges in the graph and the total number of possible edges it could have. This is a number between 0 and 1 that quantifies the amount of edges that should be added after a graph G_N with N nodes is constructed.

We now describe further parameters that the user can give as input to TØIRoads, related to the features outlined in Definition 1.

- *Road weight* (length): the user can give as input the minimum and maximum values for the cost/length of a node representing a road segment.
- *Capacity*: the user can give as input the minimum and maximum values for the capacity of a road segment. In addition, the user can specify a parameter called *capacity factor* that is used to multiply the capacity values, in this way, the user can create road data models that tend to be more or less congested, depending on how the capacity factor is defined.
- *Population*: the user can give as input the maximum value for the population associated with a node representing a road segment, with the minimum value being 0. There is also a rate parameter where the user can specify the proportion of nodes with non-zero value.
- *Points of interest*: the user can give as input the maximum value for the points of interest associated with a node representing a road segment, with the minimum value being 0. There is also a rate parameter where the user can specify the proportion of nodes with non-zero value.

For the minimum and maximum values above, TØIRoads chooses a value in the corresponding interval, using the uniform distribution, and generates the road data models.

4.2 The Algorithm for Road Data Model Generation

In addition to the specification in Section 3.2, the graph data has some constraints, motivated by how road networks are designed in practice. The tool is made to generate graphs of arbitrary size, with a few control elements. There are no self-loops and the graph is strongly connected, which makes sense considering the structure of a road network. We formally state these properties in Propositions 14 and 15.

▶ **Proposition 14.** *Every road data model returned by Algorithm 1 is strongly connected.*

Sketch. The algorithm starts creating a graph G_1 with just one node, which by definition, forms a strongly connected graph. Then, given G_i, with $i \geq 1$, the algorithm creates G_{i+1} by adding to G_i a new node n and two edges: one going from n to a random node in G_i and one coming from a random node to n. In this way, if G_i is strongly connected, the same will hold to G_{i+1}. ◀

If N is the number of nodes in a graph G_N then this procedure creates $2N - 2$ edges since at each iteration 2 edges are added and the initial graph has no edges. Although this procedure creates a strongly connected graph, its density is only $(2N - 2)/(N(N - 1))$, since a fully connected (directed) graph has $N(N - 1)$ edges. In practice, road networks can be much more dense. As mentioned in Section 4.1, TØIRoads allows the user to adjust this by including a parameter called *density*. In fact, the implementation can add a bit more variation to this, which is useful to create multiple similar but different graphs in the dataset, using a *variation flag*. If true then the density value given by the user is multiplied by a random number between 0.9 and 1.1. The choice of the edges to be added follows a uniform distribution. We state this property in Proposition 15.

▶ **Proposition 15.** *Every road data model returned by Algorithm 1 has density d, provided that $d > (2N-2)/(N(N-1))$ and the variation flag is set to false.*

Sketch. If the density parameter given by the user is lower than $(2N-2)/(N(N-1))$ then there is no change in density (since we want to ensure the graph is strongly connected). Otherwise, the algorithm adds edges randomly so as to reach the desired density (which is the density value d given as input by the user if the variation flag is false, otherwise, a value close to d). In detail, the algorithm first selects the $N^2 - 3N + 2$ possible edges that could be added (that is, not those $2N-2$ that were included in the creation of the strongly connected graph). Then, it randomly chooses a desired number among these possible edges to reach density d. ◀

The description of the algorithm for generating road data models is given by Algorithm 1, assuming for simplicity that the variation flag is set to false.

Algorithm 1 Road Data Model Generation.

```
input: number $n$ of road data models, interval $I$ for the number of nodes in each data model,
  density $d$, and intervals for road weight, capacity, population, and points of interest (Sec 4.1).
output: $n$ road data models satisfying Proposition 14 and Proposition 15
for $i = 1$ to $n$ do
  Randomly choose a number of nodes $m$ in $I$.
  Set road weight, capacity, population, and points of interest values to each node
  Create a strongly connected graph $G^i$ with $m$ nodes (as in Proposition 14)
  if density $> (2N-2)/(N^2-N)$ then
    Add random edges so as to reach the desired density $d$ (Proposition 15)
  end if
  Calculate road usage as in Definition 10
  Calculate congestion as in Definition 11
end for
return $G^1, \ldots, G^n$
```

4.3 Example Run

For the convenience of the reader, we present an example run illustrating our definitions in Section 3 and the strategy in Sections 4.1 and 4.2. We choose small numbers for conciseness. Suppose the user wants to create a simple dataset with just one road data model where the graph has 3 nodes. Assume TØIRoads creates G_1, G_2, G_3 as in Figure 1. Now suppose the user sets the following parameters for the features:

- road weight is a number between 1 and 5;
- road capacity is a value between 1 and 3;
- the capacity factor is 1;
- the rates for points of interest and population are both 0.7; and
- the maximum population and points of interest (per node representing a node segment) is 3.

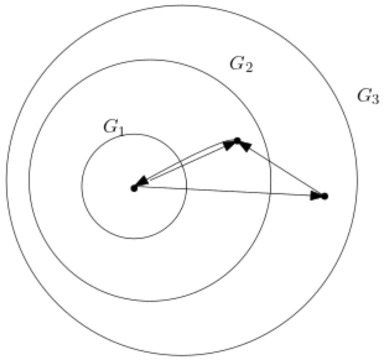

Figure 1 Graph Structure of G_1, G_2, G_3.

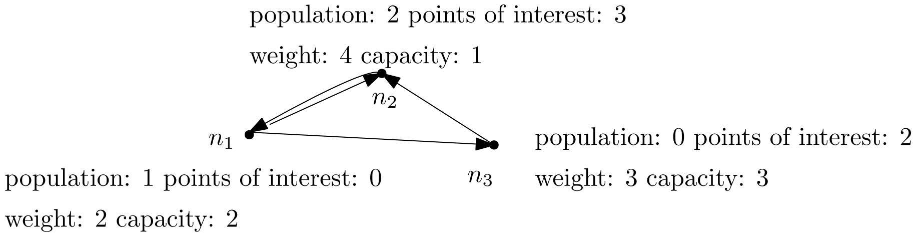
population: 2 points of interest: 3
weight: 4 capacity: 1
n_2
n_1
population: 0 points of interest: 2
population: 1 points of interest: 0
n_3
weight: 3 capacity: 3
weight: 2 capacity: 2

Figure 2 Road Data Model.

Given the parameters above and the graph structure illustrated in Figure 1, suppose TØIRoads creates the road data model illustrated in Figure 2. In the road data model of Figure 2, there are two paths from n_1 to n_2. The shortest one p_1 corresponds to the sequence n_1, n_2 (sum of weights 6) and the longest path p_2 is the sequence n_1, n_3, n_2 (sum of weights 9), by the order relation defined in Section 3.1. Considering $V = \{n_1, n_2, n_3\}$, the road usage of n_2 is

$$\sum_{(s,t)\in V^2} p_s \cdot i_t \cdot \frac{|S^{n_2}_{s,t}|}{|S_{s,t}|} = 1 \cdot 3 + 2 \cdot 3 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 3 = 9.$$

Since the capacity of this node is only 1, in this road data model, we have congestion in node n_2.

4.4 Graph Mutator

The graph mutator is an additional component added to the implementation to facilitate the creation of variants of a particular road data model given as input. The idea of this component is to give the user the possibility to train a machine learning model with enough data to simulate "what if" scenarios starting from a specific road data model, that could have realistic values from a city related to population, points of interest, etc. The parameters that the user can set are:

- the number of mutations of the original road data model that are to be generated,
- the percentage of nodes that should change
- the maximum values for the population, points of interest, and capacity.

The graph mutator component of TØIRoads chooses a value in the corresponding interval, according to the uniform distribution and generates the variations of the features values of the original graph (keeping the structure of the original graph).

5 Datasets generated by TØIRoads

In this section we provide some experimental results using TØIRoads for generating datasets. The experiments were performed on an Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz with 4 cores (x86_64 architecture) with 11 GB of memory. The code used for the experiments is available at `https://github.com/gruwesen/TOIROADS`. Our hypotheses are that (i) the run time is heavily affected by density and number of nodes, (ii) congestion is affected by the number of nodes and the maximum number of population per node, and (iii) congestion is affected by the ratio of nodes that have a population value. From the mathematical point of view, points of interest and population are interchangeable variables in S-RoadGNN. So the same behaviour for population would hold for points of interest, meaning that there would be no need to perform experiments varying the points of interest instead of population. For this reason, runtime and congestion assessments are only done varying population. We first present some results regarding the time needed to create the datasets.

Table 1 Runtime in seconds for generating 10 graphs varying the number of nodes and density.

Nodes	50	50	100	100	200	200	300	300	500	500
Density	0.04	0.5	0.04	0.5	0.04	0.5	0.04	0.5	0.04	0.5
Time	0.257	0.697	1.450	7.244	12.949	98.942	53.141	487.046	351.428	3732.413

The results in Table 1 show how increasing the number of nodes and density affects the runtime. For this experiment, the values of the remaining parameters were kept fix and set as follows: the road weight (indicating cost/length) was a number between 1 and 10, the population and points of interest rate was 0.4, the maximum for population and points of interest (per node representing a road segment) was 3, finally, road capacity was a number between 10 and 20.

We now present results showing how increasing the population (and not road capacity) can lead to more congested nodes in the road data model.

Table 2 Congestion increasing according to population increase at 0.4 population rate.

Nodes	50	50	50	100	100	100	200	200	200
Total nodes	1000	1000	1000	2000	2000	2000	4000	4000	4000
Max population	4	10	40	4	10	40	4	10	40
Congested nodes	807	898	934	1624	1681	1829	3051	3250	3428
Congestion %	80.7%	89.8%	93.4%	82.1%	84.1%	91.5%	76.3%	81.3%	85.7%

Table 2 describes how max population increases the number of congested nodes when considering three sets of 20 graphs, each with 50, 100, and 200 nodes. The capacity is set to a random number between $[3\sqrt{n}, 3n]$, where n is the number of nodes, as this interval has been found to often yield a medium amount of congestion in S-RoadGNN. As can be seen from the table, increasing the max population increases congestion at every node number.

Table 3 describes how the population rate increases the congestion level in a similar manner to Table 2. Again, we see an increase in the number of congested nodes as we increase the population rate, similar to when we increase the maximum population. To reiterate, the max population is the max value of a node that has population, while the population rate describes the part of nodes that contain a population. Decreasing the population rate to 0.2 did more to lessen congestion than decreasing the max population.

To illustrate how this dataset can be used in the context of traffic prediction, we have made an experiment using a graph neural network (GNN) that predicts node congestion. In this experiment

Table 3 Congestion increasing according to population rate increase at 40 max population.

Nodes	50	50	50	100	100	100	200	200	200
Total nodes	1000	1000	1000	2000	2000	2000	4000	4000	4000
Population rate	0.2	0.4	0.8	0.2	0.4	0.8	0.2	0.4	0.8
Congested nodes	608	778	957	1244	1575	1904	2240	2975	3749
Congestion %	60.8%	77.8%	95.7%	62.2%	78.75%	95.2%	56.0%	74.4%	93.73%

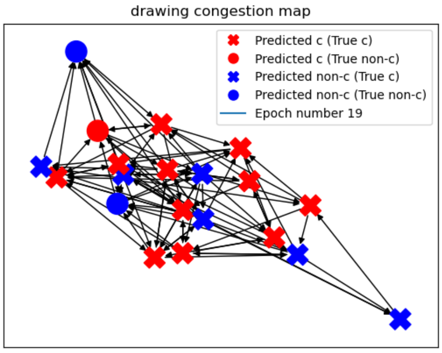
drawing congestion map
✖ Predicted c (True c)
● Predicted c (True non-c)
✖ Predicted non-c (True c)
● Predicted non-c (True non-c)
Epoch number 19

Figure 3 Example of Congestion Prediction with a Graph Neural Network.

we trained a GNN model on a dataset generated by S-RoadGNN to predict whether a node is congested or not. We included in Figure 3 a graphical representation of the prediction results of the GNN. Figure 3 shows the resulting predictions for a single example graph after twenty epochs of training. This experiment was run using a Graph Attention Network (GAT) with three layers. The training dataset contained 100 graphs of 20 nodes, while the validation set contained 40 graphs of 20 nodes. The specific graph that is drawn in Figure 3 was drawn randomly from the validation set. In this figure, the model correctly predicted congestion in 13 of the 20 nodes.

6 Conclusion

We motivate and present road data models, some of their theoretical properties and a tool, called TØIRoads, for generating datasets within these models. While one may be tempted to assume that preventing congestion is always desirable, this may not be the case if congestion of non-environmental friendly means of transportation, such as private cars, serve as incentive for the use of public transportation. Nevertheless, being able to study the scenarios related to changes in the population and traffic flow depending on points of interest and to predict congestion (whether to prevent it or not) is important for long-term city planning. As future work, we plan to include a few more parameters such as a sensible strategy for adding location information (currently only represented in a relative way, based on paths and their weights) to our road data models. A potential improvement would be to allow different densities per region of the graph, to simulate more dense districts, as commonly happens to districts that are close to the center of a city.

References

1 Azzedine Boukerche, Yanjie Tao, and Peng Sun. Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Comput. Networks*, 182:107484, 2020. doi:10.1016/j.comnet.2020.107484.

2 Azzedine Boukerche and Jiahao Wang. Machine learning-based traffic prediction models for intelligent transportation systems. *Comput. Networks*, 181:107530, 2020. doi:10.1016/j.comnet.2020.107530.

3 Khac-Hoai Nam Bui, Jiho Cho, and Hongsuk Yi. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Appl. Intell.*, 52(3):2763–2774, 2022. doi:10.1007/s10489-021-02587-w.

4 Khac-Hoai Nam Bui, Hongsuk Yi, and Jiho Cho. UVDS: A new dataset for traffic forecasting with spatial-temporal correlation. In Ngoc Thanh Nguyen, Suphamit Chittayasothorn, Dusit Niyato, and Bogdan Trawinski, editors, *Intelligent Information and Database Systems - 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, April 7-10, 2021, Proceedings*, volume 12672 of *Lecture Notes in Computer Science*, pages 66–77. Springer, 2021. doi:10.1007/978-3-030-73280-6_6.

5 Jeongwhan Choi and Noseong Park. Graph neural rough differential equations for traffic forecasting. *ACM Trans. Intell. Syst. Technol.*, 14(4):74:1–74:27, 2023. doi:10.1145/3604808.

6 Zhiyong Cui, Ruimin Ke, Ziyuan Pu, Xiaolei Ma, and Yinhai Wang. Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. *Transportation Research Part C: Emerging Technologies*, 115:102620, 2020. doi:10.1016/j.trc.2020.102620.

7 Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, Peter W. Battaglia, Vishal Gupta, Ang Li, Zhongwen Xu, Alvaro Sanchez-Gonzalez, Yujia Li, and Petar Velickovic. ETA prediction with graph neural networks in google maps. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong, editors, *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3767–3776. ACM, 2021. doi:10.1145/3459637.3481916.

8 Alessio Gravina and Danilo Numeroso. *NumGraph.* https://numgraph.readthedocs.io.

9 Xiao Han, Guojiang Shen, Xi Yang, and Xiangjie Kong. Congestion recognition for hybrid urban road systems via digraph convolutional network. *Transportation Research Part C: Emerging Technologies*, 121:102877, 2020. doi:10.1016/j.trc.2020.102877.

10 Andreas Horni, Kai Nagel, and Kay Axhausen. *Introducing MATSim*, pages 3–8. Ubiquity Press, August 2016. doi:10.5334/baw.1.

11 Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.*, 207:117921, 2022. doi:10.1016/j.eswa.2022.117921.

12 Weiwei Jiang, Jiayun Luo, Miao He, and Weixi Gu. Graph neural network for traffic forecasting: The research progress. *ISPRS Int. J. Geo Inf.*, 12(3):100, 2023. doi:10.3390/ijgi12030100.

13 Alexandra Kapp, Julia Hansmeyer, and Helena Mihaljevic. Generative models for synthetic urban mobility data: A systematic literature review. *ACM Comput. Surv.*, 56(4):93:1–93:37, 2024. doi:10.1145/3610224.

14 Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=SJiHXGWAZ.

15 Jiaqi Ma, Jiong Zhu, Yuxiao Dong, Danai Koutra, Jingrui He, Qiaozhu Mei, Anton Tsitsulin, Xingjian Zhang, and Marinka Zitnik. The 3rd workshop on graph learning benchmarks (GLB 2023). In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye, editors, *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*, pages 5870–5871. ACM, 2023. doi:10.1145/3580305.3599224.

16 Johannes Nguyen, Simon T. Powers, Neil Urquhart, Thomas Farrenkopf, and Michael Guckert. An overview of agent-based traffic simulators. *Transportation Research Interdisciplinary Perspectives*, 12:100486, 2021. doi:10.1016/j.trip.2021.100486.

17 Eren Olug, Kiymet Kaya, Resul Tugay, and Sule Gündüz Ögüdücü. IBB traffic graph data: Benchmarking and road traffic prediction model. *CoRR*, abs/2408.01016, 2024. doi:10.48550/arXiv.2408.01016.

18 Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *AAAI*, pages 914–921. AAAI Press, 2020. doi:10.1609/aaai.v34i01.5438.

19 Grunde Haraldsson Wesenberg. gruwesen/TOIROADS. Software, version 1.0., Norwegian Research Council, project 322480, swhId: swh:1:dir:a86388944844fcc00f4cad67e1ec75a998f36eae (visited on 2024-12-11). URL: https://github.com/gruwesen/TOIROADS, doi:10.4230/artifacts.22621.

20 Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3634–3640. ijcai.org, 2018. doi:10.24963/ijcai.2018/505.