

Article

Mining Topological Dependencies of Recurrent Congestion in Road Networks

Nicolas Tempelmeier ^{1,*} , Udo Feuerhake ², Oskar Wage ² and Elena Demidova ³ 

¹ L3S Research Center, Leibniz University Hannover, 30167 Hannover, Germany

² Institute of Cartography and Geoinformatics, Leibniz University Hannover, 30167 Hannover, Germany; Udo.Feuerhake@ikg.uni-hannover.de (U.F.); Oskar.Wage@ikg.uni-hannover.de (O.W.)

³ Data Science & Intelligent Systems Group (DSIS), University of Bonn, D-53012 Bonn, Germany; elena.demidova@cs.uni-bonn.de

* Correspondence: tempelmeier@L3S.de

Abstract: The discovery of spatio-temporal dependencies within urban road networks that cause Recurrent Congestion (RC) patterns is crucial for numerous real-world applications, including urban planning and the scheduling of public transportation services. While most existing studies investigate temporal patterns of RC phenomena, the influence of the road network topology on RC is often overlooked. This article proposes the ST-DISCOVERY algorithm, a novel unsupervised spatio-temporal data mining algorithm that facilitates effective data-driven discovery of RC dependencies induced by the road network topology using real-world traffic data. We factor out regularly reoccurring traffic phenomena, such as rush hours, mainly induced by the daytime, by modelling and systematically exploiting temporal traffic load outliers. We present an algorithm that first constructs connected subgraphs of the road network based on the traffic speed outliers. Second, the algorithm identifies pairs of subgraphs that indicate spatio-temporal correlations in their traffic load behaviour to identify topological dependencies within the road network. Finally, we rank the identified subgraph pairs based on the dependency score determined by our algorithm. Our experimental results demonstrate that ST-DISCOVERY can effectively reveal topological dependencies in urban road networks.

Keywords: road network analysis; recurrent congestion; spatio-temporal data mining



Citation: Tempelmeier, N.; Feuerhake, U.; Wage, O.; Demidova, E. Mining Topological Dependencies of Recurrent Congestion in Road Networks. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 248. <https://doi.org/10.3390/ijgi10040248>

Academic Editors: Wolfgang Kainz, Géraldine Del Mondo, Peng Peng, Feng Lu and Jérôme Gensel

Received: 29 January 2021

Accepted: 4 April 2021

Published: 8 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Urban road networks possess complex interdependencies that can become apparent during congestion events [1]. Established traffic research distinguishes between Recurrent Congestion (RC), e.g., rush hours, and Non-Recurrent Congestion events, e.g., accidents [2]. This article aims to identify topological dependencies within the road network that may cause RC phenomena, henceforth called *structural dependencies*. Such dependencies are often not well understood, can become apparent only under real traffic load and can cause co-occurring RC patterns in the road network. Therefore, understanding topological dependencies in urban road networks is crucial for many real-world applications, including city planning, traffic management and public transportation services.

We illustrate structural dependencies in urban road networks at the example of the area of Gehrden—a town in the district of Hanover, Germany—in Figure 1. This figure illustrates two subgraphs of the road network (marked blue and purple). Both subgraphs represent the feeder roads to the main highway (B65) connecting Gehrden and the City of Hanover that constitutes the main commuting route for the Gehrden inhabitants. During a period with a high traffic load (e.g., during a rush hour), these subgraphs are typically simultaneously congested due to the network topology. Thus, we consider such subgraphs to be structurally dependent.

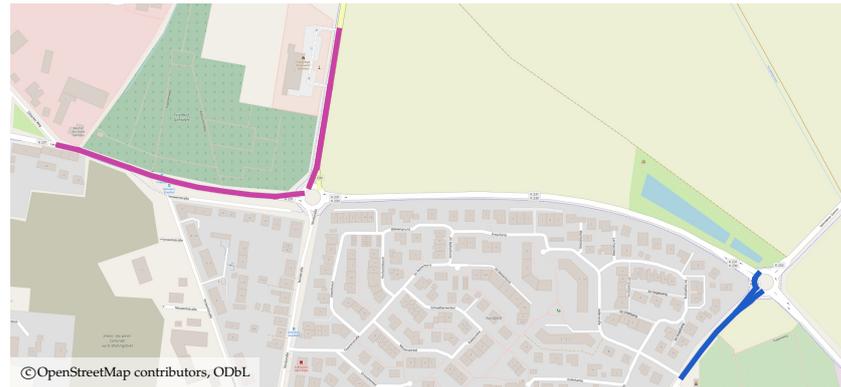


Figure 1. Example of structural dependencies in an urban road network observed near Gehrden, Germany.

The existing literature on RC mainly identifies temporal patterns [3,4]. However, we observe a lack of methods that investigate the influence of road network topology on RC. Detection of such dependencies within complex road networks is not trivial, particularly due to the variety of the influence factors (e.g., planned special events, accidents, construction sites and extreme weather conditions) and their dynamic impact on the traffic flow concerning the spatial and temporal dimensions. To the best of our knowledge, the task of data-driven discovery of structural dependencies in urban road networks is new and has not been addressed in the literature.

This article presents ST-DISCOVERY—a novel unsupervised data-driven spatio-temporal data mining algorithm to reveal structural dependencies in urban road networks. ST-DISCOVERY relies on the intuition that structural dependencies can manifest as correlations of congestion patterns. In this article, we represent congestion patterns as subgraphs of the road network. We aim to exclude RC patterns that are mainly induced by temporal factors such as rush hour patterns that mainly depend on the time of day. To this end, we only consider temporal traffic load outliers that factor out daily patterns to construct the subgraphs. We identify the subgraphs using spatial clustering of the connected network segments that indicate a high traffic load. We consider temporal correlations of subgraphs located in spatial proximity as indicators of structural dependencies in the underlying road network.

To assess the effectiveness of the proposed ST-DISCOVERY algorithm, we conducted a case study. This case study utilises two real-world traffic datasets in the regions of Hanover and Brunswick, Germany. The study results illustrate that our method can accurately detect meaningful structural dependencies in urban road networks.

In summary, our contributions are as follows. (1) We introduce the new task of data-driven discovery of structural dependencies in road networks. (2) We propose the novel ST-DISCOVERY algorithm to detect structural dependencies using traffic flow data. (3) We conduct a case study to assess the effectiveness of the proposed method.

This article extends our prior work [5] towards this direction. Compared to the works in [5], in this article, we provide a detailed explanation of the individual algorithmic steps of ST-DISCOVERY and the run-time complexity analysis. Furthermore, we present an extensive evaluation, including the manual assessment of the identified topological dependencies and a detailed investigation of the algorithm's parameter impact. A demonstration that includes visualisation of the dependencies identified by ST-DISCOVERY in an interactive traffic analytics dashboard is available in [6].

2. Related Work

This section discusses related work in the areas of congestion analysis and spatio-temporal data mining, along with related data sources.

2.1. Congestion Analysis

Existing literature on congestion patterns distinguishes between Recurrent Congestion (RC) and Non-Recurrent Congestion (NRC) events [2]. Non-Recurrent Congestion is defined as congestion that occurs because of singular events such as accidents [7–9], extreme weather conditions [10–12] or large-scale public events [13,14]. Existing research on NRC addressed a variety of problems such as delay estimation [13,14], routing adaptation [8], congestion prediction [10,12] and NRC detection and tracking [7,11,15]. Methods for NRC detection include spatio-temporal clustering [15], regression models [10] and random forests [12].

Recurrent congestion denotes the remaining congestion events. Rush hour patterns constitute the most prominent examples of RC events [16]. A large number of studies focus on the temporal analysis of RC. One line of research addresses the prediction of RC, where machine learning models such as neural networks [17–19] or Support Vector Machines [20] currently constitute the state-of-the-art. The closely related task of short-term traffic forecasting is well studied in the existing literature [21]. Models for both RC prediction and short-term traffic forecasting typically utilise periodically reoccurring traffic patterns, e.g., rush hours and day of week patterns, to facilitate predictions. Another line of research investigates the evolution of RC patterns. Current approaches typically analyse the propagation of RC within a spatial grid [3,4,22] or a road network graph [23–25].

While these studies mainly address the temporal aspects of RC, we observe a lack of research investigating the influence of road network topology on RC. This article proposes an algorithm that filters out periodic traffic patterns and identifies mutual dependencies of subgraphs within the road network.

2.2. Spatio-Temporal Data Mining

Spatio-temporal data mining algorithms address the challenge of extracting information, e.g., frequent patterns or anomalies, from large sets of spatio-temporal data. Atluri et al. provide an overview of approaches and problems in a recent survey [26].

Previous data mining approaches for road network data often aim at identifying individual important roads or junctions within the road network. The authors of [1] introduced a data-driven approach to identify the importance of individual roads within the road network by measuring the correlation of traffic load between a particular road and the whole road network. Similarly, the authors of [27] discover individual important intersections. The authors represent trips within the road network as a tripartite graph. They compute the importance of intersections with an iterative ranking algorithm. In contrast, we consider the problem of identifying pairwise dependencies in the road network.

Several studies consider outlier detection in road traffic data. In [28], anomalous traffic flow is detected by grouping road intersections via their traffic flow patterns and self-organising maps. The authors of [29] focus on detecting outliers in the traffic load by sudden changes. ST-DISCOVERY builds upon existing outlier detection methods and exploits outlier co-occurrences and mutual information to determine spatio-temporal dependencies.

2.3. Data Sources

Road traffic data are often collected from stationary sensors, GPS devices or simulations. Stationary sensors, such as induction loops permanently installed within roads, are traditional traffic data sources. Stationary sensors usually measure high-quality and consistent traffic data but lack coverage of the road network, especially in urban environments. Existing research has widely adopted the use of data from stationary sensors [8,11,15,17]. The increasing digitisation of urban traffic has led to a boost in real-world traffic data availability. In particular, floating car data (FCD) are usually collected from GPS devices. FCD enables detailed and realistic insights into the specific regions' actual traffic load. FCD has proven to be suitable data source for the analysis of both RC [4,24] and NRC [30]. Compared to data collected from stationary sensors, FCD typically covers a larger fraction of the road network but is less consistent. Simulation-based approaches (see, e.g.,

in [31,32]) utilise the features originating from the network topology and capacity and can reveal critical parts of road networks and the possible impact of incidents. However, these methods are restricted by the approximations of the underlying models that can provide only rough estimates of the real traffic flow. In this article, we rely on FCD, representing real-world traffic flow data and can provide insights into the topological dependencies that become apparent only under real traffic conditions.

3. Problem Statement and Formalisation

In this article, we address the problem of identification of the structurally dependent subgraphs in a road network. We consider subgraphs to be structurally dependent if

1. the subgraphs are located in spatial proximity,
2. the subgraphs are typically simultaneously affected by Recurrent Congestion or
3. the road network topology causes the correlation of the congestion on these subgraphs.

In the following, we formalise the key concepts adopted in the article. In this formalisation, we adopt and, where necessary, extend some of the concepts defined in our previous work [5].

Transportation graph. We represent the road network as a directed multi-graph $TG := (V, U)$, referred to as a *transportation graph*. U is a set of edges (i.e., road segments); V is a set of nodes (i.e., junctions). We refer to an edge of the transportation graph as a *unit* $u \in U$.

Unit load. We denote the traffic flow observed on a unit at a particular time point as *unit load*. Formally, $ul(u, t)$ is the traffic load on the unit u at the time point $t \in \mathcal{T}$, where \mathcal{T} denotes the set of time points.

We measure the **unit load** $ul(u, t) \in [0, 1]$ as the relative speed reduction at unit u at time point t with respect to the speed limit $lim(u)$ of the corresponding edge of the transportation graph:

$$ul(u, t) = \frac{lim(u) - speed(u, t)}{lim(u)},$$

where $speed(u, t)$ represents the traffic speed on unit u at time t .

Affected unit. We denote a unit that exhibits an abnormally high traffic load at a certain time point t as an *affected unit* at t . Formally, $affected(u, t) : U \times \mathcal{T} \mapsto \{True, False\}$ indicates whether unit u is affected at time point t .

Subgraph. We refer to a subgraph of the transportation graph as *subgraph* $sg := (V', U')$ with $V' \subset V, U' \subset U$.

Affected subgraph. An *affected subgraph* represents a subgraph that exhibits an abnormally high traffic load at a certain time point t (e.g., a congested highway section). Formally, $affected(sg, t) : SG \times \mathcal{T} \mapsto \{True, False\}$ indicates whether a subgraph sg is affected at time point t , where $SG = \{\mathcal{P}(V) \times \mathcal{P}(U)\}$ denotes the set of possible subgraphs and $\mathcal{P}(\cdot)$ denotes the power set.

We consider a subgraph sg containing the units $sg.U$ to be affected at time point t if at least one of its units $u \in sg.U$ is affected at t :

$$affected(sg, t) = \begin{cases} True, & \exists u \in sg.U : affected(u, t) \\ False, & \text{otherwise.} \end{cases}$$

4. Proposed Solution

In this article, we aim to determine structurally dependent subgraphs in a road network.

Our ST-DISCOVERY approach consists of the following main steps illustrated in Figure 2: (i) We identify affected units of the transportation graph using traffic data. (ii) We develop algorithms to identify affected subgraphs of the transportation graph. (iii) We develop an algorithm to identify structural spatio-temporal dependencies of the identified subgraphs.

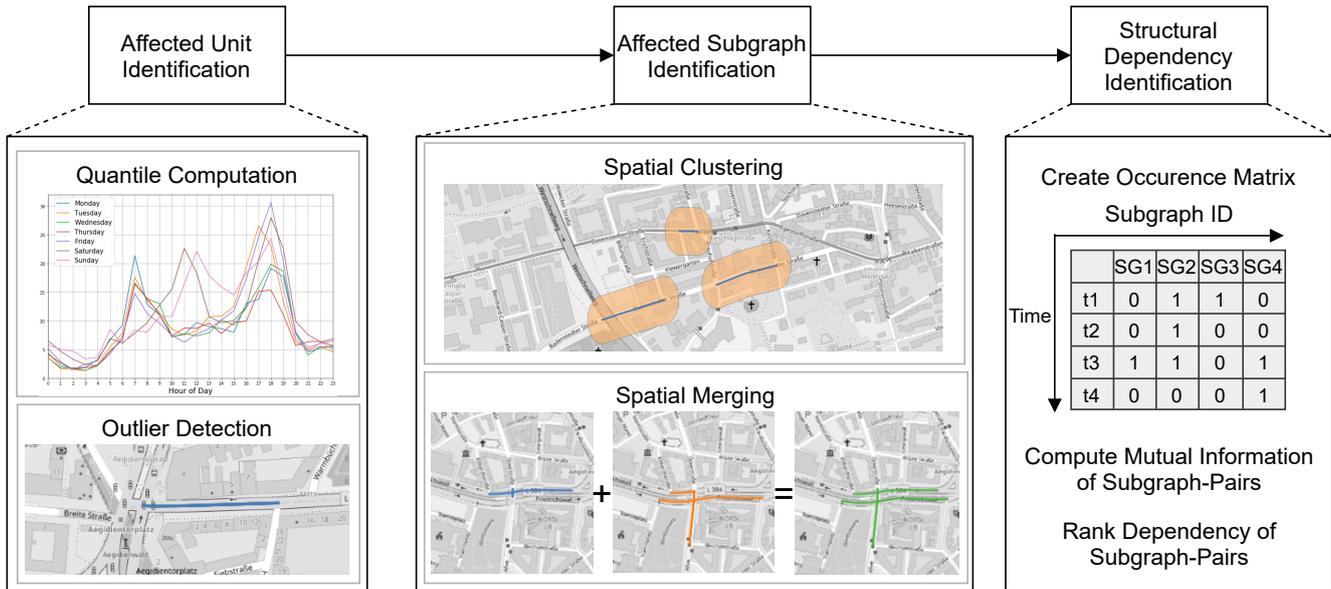


Figure 2. ST-DISCOVERY pipeline overview. Map images: ©OpenStreetMap contributors, ODbL.

4.1. Identification of Affected Units

This step aims to identify affected units, i.e., the units that exhibit an exceptionally high load at any single time point. Factors influencing road traffic include reoccurring temporal factors (such as, e.g., rush hours) and spatial factors such as road network topology. As we aim to identify Recurrent Congestion that result from the road network topology, it is desirable to exclude reoccurring temporal factors from our analysis. To exclusively factor out temporal congestion patterns, we identify affected units as temporal outliers by employing the *interquartile range* (IQR) rule [33]. As whether the day is a weekday and the time of day strongly influence the traffic load, we identify units that exhibit an exceptionally high load compared to the average traffic load on these units at the same weekday and day time. More formally, we consider u to be affected at time t , if the following condition holds:

$$affected(u, t) = \begin{cases} True, & \text{if } ul(u, t) > Q_3(u, t) + 1.5 \cdot IQR \\ False, & \text{otherwise,} \end{cases}$$

where $Q_n(u, t)$ denotes the n th quartile of the unit load on unit u regarding the weekday and day time, and $IQR = Q_3(u, t) - Q_1(u, t)$ denotes the interquartile range.

We precompute the upper bound of the unit load $Q_3(u, t) + 1.5 \cdot IQR$ for each unit, weekday and day time. We consider each unit u that exhibits a higher unit load $ul(u, t)$ than the corresponding upper bound to be affected at time point t .

4.2. Identification of Affected Subgraphs

This step aims to identify disjunctive subgraphs of the transportation graph. The units of an individual subgraph should exhibit the same unit load pattern for a given point in time. We approach this goal by conducting spatial clustering of the transportation graph's affected units. In this step, clustering is performed independently at each point in time. To ensure the spatial continuity of the resulting subgraphs, the clustering of affected units follows the basic region growing principle [34].

We conduct the clustering as follows. First, we put one seed point on each affected unit. During the next steps, the regions are expanded by merging neighbored regions until there is no further change. The neighbourhood of two regions is determined by evaluating the distance between their closest edges in the transportation graph. This distance is measured as the edge count d_u within the shortest path between the regions.

As the data can potentially be incomplete or contain measurement errors, we allow for certain tolerance while determining the neighbourhood. To this extent, we introduce the threshold $d_{u,max}$ to bridge gaps of a predefined size between two regions. Thus, two regions are considered as neighboured and are merged by the algorithm if the condition $d_u \leq d_{u,max}$ holds. As a result, the units affected at time point t are clustered into a set of n clusters $C_t = \{c_0, \dots, c_n\}$.

Note that this clustering approach utilises the transportation graph's underlying graph structure. Thus, there is a unique mapping between each cluster and the corresponding subgraph of TG . In the following, while referring to the clustering results, we use the terms cluster and affected subgraph interchangeably.

4.3. Spatial Merging of Affected Subgraphs

The affected subgraphs identified in Section 4.2 are spatially disjoint with respect to the specific time points. Intuitively, when considering the traffic load on the transportation graph over a longer time, we can observe spatial variations of the affected subgraphs, e.g., due to the congestion propagation along the graph. Furthermore, affected subgraphs (and their variations) can reoccur at different points in time. To capture these patterns, we conduct a merging of spatially overlapping affected subgraphs that occur at different time points.

Algorithm 1 presents an incremental greedy approach to merge spatially overlapping affected subgraphs. The algorithm consist of a main loop (line 6–24) where the individual steps include candidate generation (line 9–11), similarity computation (line 12–14) and merging (line 15–24). For the *candidate generation*, we consider all subgraph pairs that share at least one unit as candidates (line 13). Here, $[\mathcal{P}(\cdot)]^2$ denotes the subset of the power set with elements of cardinality 2.

Similarity computation is performed for all candidates (i.e., subgraph pairs) by computing the similarity function $\text{similarity} : SG \times SG \mapsto [0, 1]$ as follows (line 14):

$$\text{similarity}(sg_1, sg_2) = \begin{cases} 1 & \text{if } sg_1 \subset sg_2 \vee sg_2 \subset sg_1, \\ \frac{|sg_1 \cap sg_2|}{|sg_1 \cup sg_2|}, & \text{otherwise.} \end{cases}$$

Based on the definition of the affected subgraph, we consider the subgraph pairs in which one subgraph entirely contains the other subgraph as a special case that has the maximum similarity of 1. Otherwise, the similarity is computed as the *Jaccard similarity*, which measures to which extent the subgraph units overlap.

Finally, the *merging* step aggregates the subgraph pairs with a similarity score above the threshold th_{sim} (line 19–23). The pairs with the highest similarity are merged first (line 16). Here, the function `ordered()` orders the subgraph pairs in the descending similarity order. As merging affects the similarity computation, in each iteration of the algorithm, any subgraph can be merged only once (line 17–18).

The run time complexity of Algorithm 1 arises from combination of the main while loop in line 7 ($\mathcal{O}(|C|)$) and the iteration over the ordered set of subgraph pairs in line 16 ($\mathcal{O}(|C|^2 \cdot \log(|C|^2)) = \mathcal{O}(|C|^2 \cdot \log(|C|))$), resulting in a total complexity of $\mathcal{O}(|C|^3 \cdot \log(|C|))$.

To facilitate an efficient candidate generation, we maintain a hashmap (*commonUnits*) that provides a mapping from a single unit u to all subgraphs that contain this unit u (lines 2–5). The computation of all subgraph combinations would require quadratic time ($\mathcal{O}(|C|^2)$) in each iteration. In contrast, the hashmap is computed once ($\mathcal{O}(|C| \cdot |U|)$) and is then updated iteratively during the algorithm according to the performed merging using the function `update()` (line 23). The loops in lines 10 and 13 can be executed in parallel for further optimisation.

Algorithm 1 Merge Subgraphs

```

Input:   C:   Set of subgraphs
Output:  SG:  Set of merged subgraphs

1: SG ← C
2: commonUnits ← []
3: for all sg ∈ SG do
4:   for all u ∈ sg.U do
5:     commonUnits[u] ← commonUnits[u] ∪ sg
6: changed ← True
7: while changed do
8:   changed ← False
9:   {Generate candidates}
10:  candidates ← ∅
11:  for all u ∈ commonUnits do
12:    candidates ← candidates ∪ [P(commonUnits[u])]²
13:    {Compute similarities}
14:  s[] ← ∅
15:  for all (sg1, sg2) ∈ candidates do
16:    s[(sg1, sg2)] ← similarity(sg1, sg2)
17:    {Merge subgraphs}
18:  visited ← ∅
19:  for all (sg1, sg2) ∈ ordered(s) do
20:    if sg1 ∈ visited ∨ sg2 ∈ visited then
21:      continue
22:    if s[(sg1, sg2)] ≥ thsim then
23:      sg1 ← sg1 ∪ sg2
24:      SG ← SG \ sg2
25:      visited ← visited ∪ {sg1, sg2}
26:      update(commonUnits)
27:      changed ← True
28: return SG

```

4.4. Identification of Spatio-Temporal Dependencies

In this step, we aim to identify dependent subgraphs of the transportation graph, i.e., the subgraphs that are typically simultaneously affected and are located in spatial proximity. To this extent, we consider the subgraphs identified and merged in Section 4.3. These subgraphs represent topologically connected subgraphs of the transportation graph, including their spatial variations, which have been affected at some point(s) in time. In this step, we bring the temporal dimension into consideration and aim to identify the pairs of these subgraphs that are typically simultaneously affected.

Algorithm 2 presents the method to identify such subgraph pairs, where the individual steps include candidate generation (lines 8–15), score computation (lines 16–19) and sorting (line 20).

First, an occurrence matrix $occ[][]$ including the subgraphs and the time points is computed (lines 1–7), where the columns correspond to the subgraphs and the rows to the time points. If a subgraph is affected at time point t , then the corresponding cell is set to 1; otherwise, to 0. From the occurrence matrix, candidate subgraph pairs are generated (lines 8–15). Each subgraph pair that is affected simultaneously in at least one time point is considered as a candidate pair. For each candidate pair, we compute the subgraph dependency score. The intuition behind this score is to capture both the temporal

co-occurrence and the spatial proximity of the subgraphs. Therefore, the score is computed as a combination of the mutual information and an inverse spatial distance metric:

$$\text{dependency}(sg_1, sg_2) = \begin{cases} 0, & \text{if } dist(sg_1, sg_2) \leq dist_{min} \\ mi(sg_1, sg_2) \cdot \frac{1}{dist(sg_1, sg_2)}, & \text{otherwise.} \end{cases}$$

Here, $dist(sg_1, sg_2)$ denotes the shortest geographic distance between two subgraphs. The threshold $dist_{min}$ specifies the minimum geographic distance for a subgraph pair to be considered dependent. $dist_{min}$ allows excluding trivial dependencies, such as adjacent subgraphs. The mutual information $mi(sg_1, sg_2)$ aims to assess the temporal co-occurrence of two subgraphs, computed as

$$mi(sg_1, sg_2) = \sum_{t_1 \in \mathcal{T}_1} \sum_{t_2 \in \mathcal{T}_2} p_{(t_1, t_2)}(t_1, t_2) \log \left(\frac{p_{(t_1, t_2)}(t_1, t_2)}{p_{t_1}(t_1) p_{t_2}(t_2)} \right),$$

where $\mathcal{T}_i = \{t \in \mathcal{T} \mid affected(sg_i, t)\}$ denotes the set of time points in which the subgraph sg_i is affected. The spatial proximity is measured as the inverse distance, where $dist(sg_1, sg_2)$ denotes the shortest geographic distance between two subgraphs. Finally, the subgraph pairs are ordered in the descending order of their dependency scores (line 20, `ordered()`).

The run time complexity of Algorithm 2 results from the identification of candidates ($\mathcal{O}(|SG|^2 \cdot \mathcal{T})$) in line 9 and line 12 as well as the sorting of subgraph pairs by score ($\mathcal{O}(|SG|^2 \cdot \log(|SG|))$) in line 20. Therefore, the overall complexity is bounded by $\mathcal{O}(|SG|^2 \cdot (\mathcal{T} + \log(SG)))$. Finally, the for loop in line 17 can be executed in parallel.

Algorithm 2 Determine Spatio-Temporal Subgraph Dependencies

Input: SG : Set of subgraphs
 \mathcal{T} : Set of time points
Output: $P_{dependent}$ Set of pairs of subgraphs,
ordered by dependency score

```

1:  $occ[\ ][\ ] \leftarrow \emptyset$ 
2: for all  $t \in \mathcal{T}$  do
3:   for all  $sg \in SG$  do
4:     if  $\exists u \in SG : iqr(u, t)$  then
5:        $occ[t][sg] \leftarrow 1$ 
6:     else
7:        $occ[t][sg] \leftarrow 0$ 
      {Determine candidate pairs}
8:  $candidates \leftarrow \emptyset$ 
9: for all  $(sg_1, sg_2) \in [P(SG)]^2$  do
10:  if  $(sg_1, sg_2) \in candidates$  then
11:    continue
12:  for all  $t \in \mathcal{T}$  do
13:    if  $occ[t][sg_1] = 1 \wedge occ[t][sg_2] = 1$  then
14:       $candidates \leftarrow candidates \cup \{(sg_1, sg_2)\}$ 
15:    break
      {Compute dependency}
16:  $P_{dependent} \leftarrow \ []$ 
17: for all  $(sg_1, sg_2) \in candidates$  do
18:    $score \leftarrow dependency(sg_1, sg_2, \mathcal{T})$ 
19:    $P_{dependent}[(sg_1, sg_2)] \leftarrow score$ 
20: return  $ordered(P_{dependent})$ 

```

We provide an open-source implementation of the ST-DISCOVERY algorithms under the MIT-license (<https://github.com/Data4UrbanMobility/st-discovery>, accessed on 19 March 2021).

5. Datasets

5.1. OpenStreetMap

OpenStreetMap (OSM) (<https://www.openstreetmap.org>, accessed on 19 March 2021) is a provider of publicly available map data. We make use of the OSM road network to form the transportation graph TG . OSM partitions roads in smaller road segments that correspond to the transportation graph units TG . In particular, we extract the road segments located within the cities of Hanover and Brunswick, Germany. Considering the OSM-taxonomy for road types, we restrict the transportation graph to the major roads, as reliable traffic information for smaller roads is rarely available. In particular, we extract all roads that belong to one of the following classes: {primary, primary_link, secondary, secondary_link, tertiary, tertiary_link, motorway, motorway_link, trunk, trunk_link}. The extracted transportation graphs contain approximately 23,000 units (Hanover) and 7600 units (Brunswick) in total. For each unit $u \in TG$, the available information regarding the speed limit $lim(u)$ as well as the road type is extracted from OSM.

5.2. Traffic Dataset

The experiments conducted in this article employ a proprietary traffic dataset that contains aggregated floating car data. In particular, the dataset provides traffic speed records for each unit u of the transportation graph TG . The dataset was collected by a company offering routing software. The dataset contains data contributions obtained from various sources, including the data collected from the users of the routing software and traffic data acquired from third-party data providers. Although detailed statistics of these contributions, such as the number or the types of the monitored vehicles, are not available to the authors due to the variety of sources involved, we do not expect any particular biases towards certain vehicle types or expense classes.

The traffic data records contain the average traffic speed on the individual transportation graph units at discrete time points, i.e., $speed(u, t)$, recorded every 15 min. The average speed records are computed by the data provider by calculating the average traffic speed from the raw floating car data, averaged over all vehicles for which the data are available for the given unit and time interval. In the dataset, only the aggregated traffic speed information $speed(u, t)$ is available. The data for the major road categories mentioned in Section 5.1 are captured regularly within the dataset. Table 1 presents statistics about the number of available traffic data records for Hanover and Brunswick. We believe that the available data are sufficient to capture typical congestion patterns.

Table 1. Dataset statistics for Hanover and Brunswick.

	Hanover	Brunswick
No. Units	23,125	7678
No. Records	195×10^6	43×10^6
Avg. No. Records/Unit	8422.79	5674.91
Time Span	October 2017–January 2018	December 2018–January 2019

6. Experiments and Discussion

The evaluation aims to assess the effectiveness of the proposed ST-DISCOVERY approach and its applicability to the real-world datasets presented in Section 5. First, we present an assessment of the dependencies identified by ST-DISCOVERY. Second, we evaluate and discuss the results of each main step of ST-DISCOVERY, i.e., the identification of affected units and subgraphs, and the merging subgraphs.

6.1. Structural Dependencies

The task of data-driven identification of structural dependencies between subgraphs of a road network is new, such that neither a baseline nor any gold standard exists. Therefore, to assess the quality of the identified structural dependencies, we conduct a manual evaluation. In this evaluation, we use ST-DISCOVERY to generate a ranked list of top- k subgraph pairs with high dependency scores, while using different values of th_{sim} . To exclude trivial dependencies, i.e., the subgraphs that are adjacent to each other, we set the threshold $dist_{min} = 500$ meters. We set $d_{u,max} = 1$ for Hanover and $d_{u,max} = 2$ for Brunswick.

We manually assess the correctness of each of the top- k subgraph pairs with the highest dependency scores. We judge each pair to be correct if we can observe and explain a structural dependency, or incorrect otherwise. The article authors performed the evaluation, while we discussed the individual judgements to obtain consensus. Finally, we calculate the $precision@k$ as the proportion of the results judged as correct within the top- k subgraph pairs.

Figure 3 presents the $precision@k$ for $th_{sim} \in \{0, 0.1, 0.2, 0.3\}$ for Hanover (Figure 3a) and Brunswick (Figure 3b). For both datasets, the highest $precision@k$ is achieved at $k = 10$ for all values of th_{sim} except $th_{sim} = 0$. As the k value increases (i.e., we consider more subgraph pairs with lower scores), the precision decreases in the majority of configurations. This behaviour can be expected, as the pairs with lower scores possess lower mutual information or are located at a higher geographic distance and are therefore less related. We conclude that the proposed score is well suited to quantifying the dependency of the affected subgraphs.

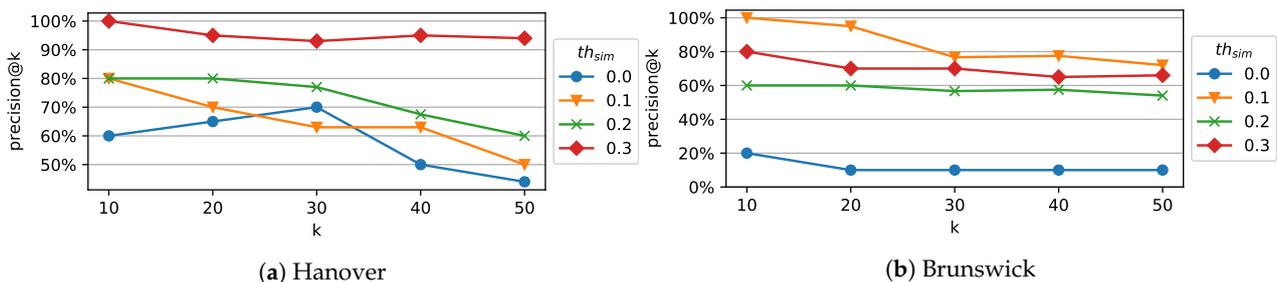


Figure 3. Precision@k with respect to k and th_{sim} of the identified structural subgraph dependencies.

The best precision at $k=10$ is achieved by $th_{sim} = 0.3$ (Hanover) and $th_{sim} = 0.1$ (Brunswick). This indicates that the optimal value of th_{sim} is dependent on the target road network. In both cases, the worst performance is achieved at $th_{sim} = 0$. The graph partitioning at $th_{sim} = 0$ is relatively coarse such that units that exhibit different dependencies can be aggregated into the same subgraph. Therefore, the achieved performance is lower than for the higher threshold values.

Note that the adopted evaluation method assesses the subgraph pairs' dependency but not the subgraphs' granularity. Partitioning with higher threshold values (i.e., $th_{sim} = 0.3$) leads to fine granular subgraphs. In this case, the partitioning can lead to a split of subgraphs that exhibit the same structural dependencies into different subgraphs, which may potentially lead to the inclusion of redundant subgraph pairs in the top- k results.

Therefore, whereas the combination of lower values of k and higher values of $th_{sim} = 0.3$ leads to the highest $precision@k$ values, it can also lead to some redundancy in the top- k results (i.e., subgraph pairs representing the same dependency at different levels of granularity). After manual inspection of the result granularity in our dataset, we observe that values of $th_{sim} \in \{0.1, 0.2\}$ lead to good results, with a precision of 80% (Hanover) and 100% (Brunswick) at $k = 10$, while avoiding redundant subgraphs in the top results. In general, we recommend that the th_{sim} threshold should be adjusted according to the specific dataset and the use case under consideration.

To facilitate a better understanding of the determined dependencies, we discuss exemplary cases identified by ST-DISCOVERY. Figure 4 provides examples of the identified dependent subgraph pairs in Hanover, where the corresponding subgraphs in a pair are marked in blue and purple, correspondingly. Figure 4a shows two junctions of a major rural street. The affected subgraph in the west includes the junction and its feeder roads, whereas only the junction is affected in the east. This combination can be caused by the drivers who avoid the larger congestion in the west by accessing the street in the east, leading to increased traffic on both junctions. Figure 4b depicts two affected subgraphs that cross the central railway within the city of Hanover. The railway divides two city districts and needs to be crossed when travelling between these districts. Therefore, the subgraphs represent alternative routes for trips from the north to the south and form a bottleneck for such trips. Figure 4c illustrates an affected highway (blue) and the last possible exit before that section (purple). If the highway is affected, the nearby exit and the consecutive roads are affected as well. This is likely caused by drivers trying to exit the highway before entering the congested part, leading to an increased load in that region. Figure 4d depicts a road leading to a town (purple) and another road leaving the town (blue). This indicates a large amount of traffic unnecessarily passing through the town transiting from the purple to the blue subgraph because of a lack of alternative routes. In this case, building a new alternative road could prevent the town from being exposed to the high traffic load.



Figure 4. Examples of the identified dependencies between subgraphs in Hanover. Dependent subgraphs are marked in blue and purple. Map images: ©OpenStreetMap contributors, ODbL.

Figure 5 provides examples of identified dependent subgraph pairs in Brunswick. Figure 5a shows two near-parallel road sections that connect similar city districts. The roads constitute alternative routes for trips to the east or west. If one of the road segments faces congestion, the other road is likely to face increased load caused by drivers that want to avoid congestion. Figure 5b presents two dependent subgraphs near the city centre of Brunswick. The subgraphs are prominent options for leaving or entering the city centre. If the city centre is congested, the congestion is likely to propagate to the subgraphs as well. Furthermore, if one subgraph is congested, the other subgraph represents an alternative route for a similar trip.

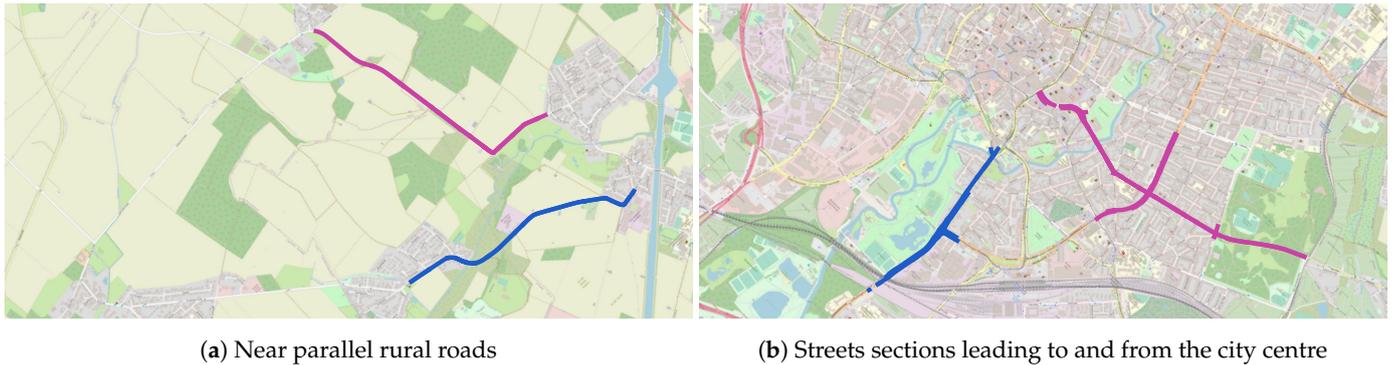


Figure 5. Examples of identified dependencies between subgraphs in Brunswick. Dependent subgraphs are marked in blue and purple. Map images: ©OpenStreetMap contributors, ODbL.

6.2. Analysis of Affected Units and Subgraphs

In this section, we analyse the distribution of the affected units and subgraphs identified by ST-DISCOVERY in our datasets and the influence of the corresponding parameter.

6.2.1. Distribution of Affected Units

The analysis results of the distribution of affected units identified by the algorithm presented in Section 4.1 is shown in Figure 6. Figure 6a presents the number of affected units per time point between November and December 2017 in Hanover. We observe that the number of affected units varies continuously, from 3 to 7724 in our dataset, with a median value of 409. Furthermore, we can observe several peaks (i.e., time points that exhibit an exceptionally high number of affected units). We observe the highest peaks between 10 December 2017 and 15 December 2017. Through a manual investigation (i.e., search for news articles related to this region and dates on Google), we found that heavy snowfall caused high and unusual delays on the whole road network during this period. This is reflected in a high number of affected units throughout the entire network. Figure 6b presents the number of affected units per time point between December 2018 and January 2019 in Brunswick. We observe a similar continuous variation of the number of affected units from 59 to 770 with a median value of 203.

Smaller peaks occur at several times, for instance, at 3 December 2017 (Hanover) and 26 January 2019 (Brunswick), where the size of the peaks highly differs. Given these observations, we believe that temporary peaks in the number of affected units can indicate occurrences of extraordinary incidents. Note that these observations are solely based on the temporal co-occurrences and do not provide any insights into the incidents' spatial characteristics.

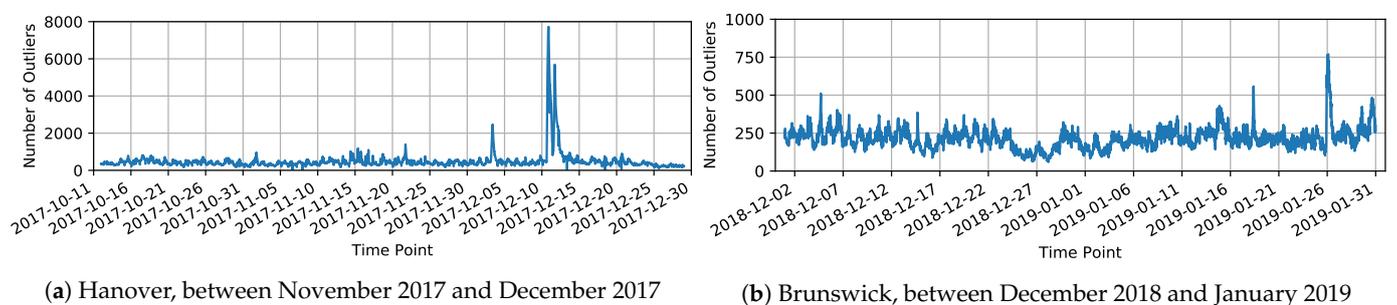


Figure 6. Number of affected units per time point.

6.2.2. Distribution of Affected Subgraphs

In this section, we analyse the distribution of the affected subgraphs identified using the algorithms presented in Section 4.2 and the influence of the relevant parameters.

During the identification of affected subgraphs using clustering of affected units presented in Section 4.2, the threshold $d_{u,max}$ was introduced, which describes the distance tolerance in assigning affected units to a subgraph. Therefore, variations in the value of this threshold lead to a different number and size of unit clusters, as illustrated in Figure 7a for Hanover and in Figure 7b for Brunswick. We evaluate values of $d_{u,max} \in [0, 100]$ to capture potential convergence patterns in the number of affected subgraphs and the average subgraph size. With an increasing tolerance (i.e., the value of $d_{u,max}$), two trends can be observed in both road networks. While the average size of the resulting subgraphs increases, their number decreases. This is caused by the fact that increasingly more subgraphs are merged if the tolerance for determining a segment neighbourhood is increased.

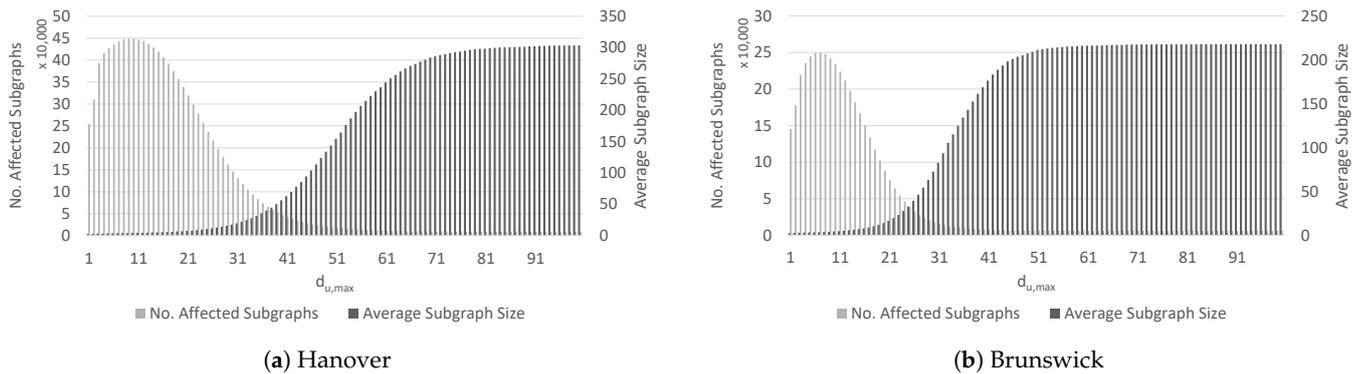


Figure 7. The number and the average size of the identified affected subgraphs show opposite trends dependent on the chosen threshold $d_{u,max}$. Subgraph size is measured as the number of edges.

Comparing the road networks with each other, we observe that the saturation of the subgraph size is reached at smaller values of $d_{u,max}$ for Brunswick (at 51) than for Hanover (at 71), as the road network of Brunswick (7678 units) is smaller than the road network of Hanover (23,125 units). At the saturation point, the entire road network is captured by a single subgraph. Therefore, a too high value of $d_{u,max}$ is not suitable to identify meaningful dependencies.

In general, the choice of a suitable value for $d_{u,max}$ strongly depends on the application scenario, the road network and the scale of the analysis. For instance, if a large region, e.g., a whole city, or one of its districts, has to be analysed; tiny subgraphs consisting of just a few road segments are not that important. For this case, a higher threshold value should be chosen. In contrast, for a detailed inspection of smaller parts of the road network, e.g., specific roads or junctions, finer subgraphs are more critical. In this use case, to prevent the merging of smaller subgraphs, a lower value of $d_{u,max}$ should be selected. Furthermore, the granularity of the road network segmentation plays an important role in the choice of $d_{u,max}$. For a fine-granular segmentation, a higher value of $d_{u,max}$ can be used to obtain larger and more meaningful subgraphs. In addition, the parameter allows addressing data sparsity by skipping segments for which no data is available. For a road network with a coarse segmentation, lower values of $d_{u,max}$ prevent the subgraphs to cover a too large spatial area.

In our experimental setting, we set $d_{u,max} = 1$ for Hanover and $d_{u,max} = 2$ for Brunswick. We observed by manual evaluation that higher values of $d_{u,max}$ on these datasets result in subgraphs that are too large to be meaningful, while the possibility to skip one or two edges leads to stable results.

6.2.3. Temporal Persistence of Affected Subgraphs

Besides the size of the identified affected subgraphs, we analyse their temporal persistence at the example of Hanover. This means that the affected subgraphs identified at time point i (including the typical variations of these subgraphs, e.g., due to the propagation of the traffic load along the transportation graph) have to be recognised in the subsequent

time steps. For this purpose, we apply an algorithm based on the Hungarian method [35]. This algorithm aims to find an optimal assignment of clusters (i.e., affected subgraphs) in two consecutive time steps by minimising the assignment costs. In order to compute the assignment costs, the intersection of the affected units involved in each subgraph (cluster) in two subsequent time steps i and j is calculated as

$$\text{costs}_{i,j} = ||c_i \cap c_j|| \quad \forall c_i \in C_i, c_j \in C_j.$$

Further, no tolerance would allow a subgraph to skip the subsequent time steps. This means the subgraph existence time will not be prolonged if this subgraph does not appear in a time step. Thus, if there is no assignment for a current subgraph in the following time step, this subgraph will “die”. If there is a cluster of affected units located on the same road segments in the next but one time step, this cluster will be treated as a new affected subgraph, i.e., a new identifier will be assigned to this subgraph and a new existence time will be initiated.

As we assume that the existence time of a cluster depends on its size, we evaluated the subgraph existence time in dependency of their size. As discussed in Section 6.2.2, the size of the subgraphs depends on $d_{u,max}$. Analogous to the study on subgraph sizes, we investigate the values of $d_{u,max} \in [0, 100]$. Figure 8 shows the median existence time of the clusters in relation to the average subgraph size and the chosen threshold $d_{u,max}$. The overall trend indicates that the affected subgraphs’ existence time increases with their size. The median existence time for lower values of $d_{u,max} \in [0, 40]$ ranges from 15 to 90 min. These existence times correspond to typical traffic congestion durations. For higher values of $d_{u,max} \in [71, 100]$, a single subgraph covers the whole road network. This results in extreme existence times of up to 10^5 minutes such that the persistence of the affected subgraph covers the whole dataset. Thus, the choice of $d_{u,max}$ does not only influence the subgraph size, but also its existence time.

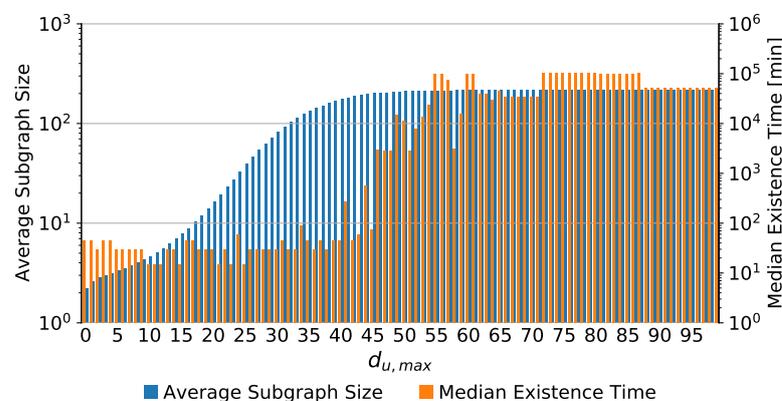


Figure 8. The existence time and the size of the affected subgraphs dependent on the value of the $d_{u,max}$ threshold. The axes are in logarithmic scale.

6.3. Evaluation of Subgraph Merging

In this section, we analyse the influence of the threshold th_{sim} on the results of the subgraph merging conducted using Algorithm 1. Figure 9 opposes the number of subgraphs and the average size of subgraphs computed by Algorithm 1 with respect to the similarity threshold th_{sim} that specifies to which fraction two subgraphs need to overlap to be merged. Higher threshold values are more restrictive.

In general, as th_{sim} increases, we observe a growing number of subgraphs, whereas the average size of these subgraphs decreases. This indicates that higher th_{sim} values result in a finer granular division of the road network into smaller subgraphs. The highest change of the average subgraph size can be observed for $th_{sim} \in [0, 0.4]$. For $th_{sim} > 0.4$, we only observe small changes of the average subgraph size. We conclude that values within $[0, 0.4]$ are particularly suited to calibrate the algorithm in the considered setup, whereas higher

threshold values result in a larger number of subgraphs (up to 50 k independent subgraphs in Hanover) and only weakly affect the subgraph size.

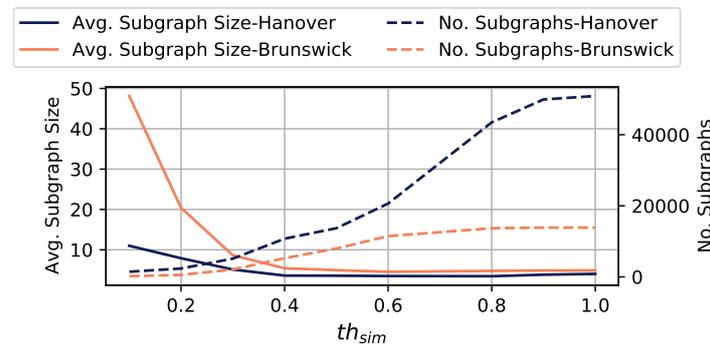


Figure 9. Influence of th_{sim} in Algorithm 1 on the number of subgraphs and an average subgraph size.

We illustrate the influence of th_{sim} at the example of a major junction in Hanover within our dataset. Figure 10 presents six different partitionings of the road network into subgraphs with respect to the values of $th_{sim} \in [0, 0.5]$, where the colour of the units represents their assignment to different subgraphs. In general, the subgraphs become finer granular with an increasing value of th_{sim} . For $th_{sim} = 0$ (the least restrictive value), we observe that all units in the considered area are assigned to a single subgraph. Note, that Algorithm 1 with $th_{sim} = 0$ will not automatically merge all subgraphs, but only those who share at least one common unit. For $th_{sim} = 0.1$, the junction is partitioned into three major subgraphs corresponding to the north (green), south (yellow) and west (purple) part of the network. Further increase of th_{sim} leads to finer granular partitions of the junction. For instance, for $th_{sim} = 0.3$ individual subgraphs for roads in the northwest (pink) and the northeast (purple) are present. Finally, for $th_{sim} \in \{0.4, 0.5\}$ we observe a fine granular partitioning of the junction in a large number of subgraphs, where individual subgraphs may contain only a few units. This corresponds to the increase in the number of subgraphs in Figure 9 for high values of th_{sim} .

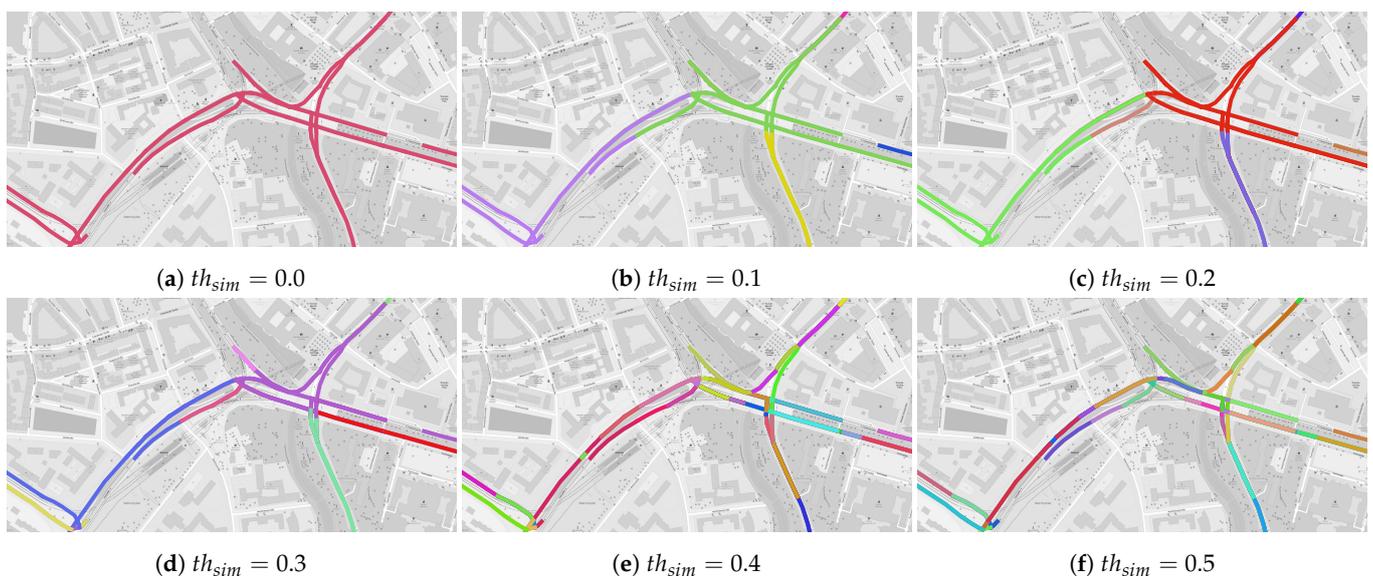


Figure 10. Example of affected subgraphs calculated for a major junction in Hanover with varying values of th_{sim} . Colours of the road segments indicate the subgraph assignment of the segments. Map images: ©OpenStreetMap contributors, ODbL.

Overall, th_{sim} can be used to adjust the granularity of ST-DISCOVERY. Whereas lower threshold values result in large subgraphs covering larger fractions of the road network

($th_{sim} = 0$), subgraphs obtained using higher threshold values (e.g., $th_{sim} = 0.4$) cover much smaller groups of affected units.

Similar to the other parameters, the concrete choice of th_{sim} depends on the use case and the road network. In the example of Figure 10, the threshold can be set to $th_{sim} = 0$ to investigate dependencies of the junction with, for instance, other junctions. Higher values, e.g., $th_{sim} = 0.2$, can differentiate between the cardinal directions of the junction, i.e., West (Green), Northeast (Red) and South (Purple). Increasing the threshold further to $th_{sim} = 0.4$ enables the discrimination between individual lanes of the roads.

In our experimental setting, we evaluated the values $th_{sim} \in \{0.0, 0.1, 0.2, 0.3\}$, where we obtained the most meaningful results with $th_{sim} = 0.3$ for Hanover and $th_{sim} = 0.1$ for Brunswick.

7. Conclusions and Outlook

In this article, we addressed the problem of data-driven discovery of topological dependencies of Recurrent Congestion within urban road networks. We presented the ST-DISCOVERY approach—a novel method to detect such dependencies based on traffic outlier analysis. ST-DISCOVERY detects the units (i.e., road segments) within the road network that indicate an exceptionally high traffic load, proposes algorithms to identify affected subgraphs within the road network using these units and identifies spatio-temporal dependencies among these subgraphs. Furthermore, ST-DISCOVERY provides parameters to adjust the granularity of the identified subgraphs to specific use cases. Our evaluation results on the real-world datasets demonstrate that ST-DISCOVERY can detect meaningful spatio-temporal dependencies among the subgraphs in urban road networks. The identified RC patterns include, for example, dependencies in the feeder roads of highways; alternative routes in case of traffic disruptions; or typical routes to POIs such as, e.g., event venues.

We believe that our approach is applicable in a variety of contexts, such as mining topological dependencies in the presence of events and accidents, and under different weather conditions. In this article, we considered a dataset that covers a single season, i.e., winter. If seasonal patterns are of interest, each season can be investigated separately using our method by partitioning the data with respect to the season. Another interesting addition to our approach would be to use of further information, for instance, the number of vehicles or the vehicle types. Furthermore, we envision that our algorithm can be used to optimise diversion routing in the presence of temporary road obstacles such as construction sites or traffic incidents. The applicability and usefulness of the approach in these contexts and use cases are subjects for future research. Finally, we intend to address the aspects of explainability of ST-DISCOVERY results for end users, including, e.g., city planners and traffic managers.

Author Contributions: Conceptualization, Nicolas Tempelmeier, Udo Feuerhake, Oskar Wage, and Elena Demidova; methodology, Nicolas Tempelmeier, Udo Feuerhake, Oskar Wage, and Elena Demidova; software, Nicolas Tempelmeier, and Udo Feuerhake; validation, Nicolas Tempelmeier, Udo Feuerhake, and Oskar Wage; formal analysis, Nicolas Tempelmeier, and Udo Feuerhake; investigation, Nicolas Tempelmeier, Udo Feuerhake, and Oskar Wage; data curation, Nicolas Tempelmeier, Udo Feuerhake, and Oskar Wage; writing—original draft preparation, Nicolas Tempelmeier, Udo Feuerhake, Oskar Wage, and Elena Demidova; writing—review and editing, Nicolas Tempelmeier, Udo Feuerhake, and Elena Demidova; visualization, Nicolas Tempelmeier, and Udo Feuerhake; supervision, Elena Demidova; funding acquisition, Elena Demidova. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially funded by the BMBF and the BMWi, Germany under the projects “Data4UrbanMobility” (grant ID 02K15A040), “USEFUL” (grant ID 03SF0547), “CampaNeo” (grant ID 01MD19007B), “d-E-mand” (grant ID 01ME19009B), the European Commission (EU H2020, “smashHit”, grant-ID 871477) as well as by the research initiatives “Mobiler Mensch” and “Urbane Logistik Hannover”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data utilised in this article cannot be shared due to legal restrictions. We provide an open-source implementation of the ST-DISCOVERY algorithms under the MIT-license (<https://github.com/Data4UrbanMobility/st-discovery>, accessed on 19 March 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, S.; Zhou, D.; Fan, J.; Tong, Q.; Zhu, T.; Lv, W.; Li, D.; Havlin, S. Identifying the most influential roads based on traffic correlation networks. *EPJ Data Sci.* **2019**, *8*, 1–17. [[CrossRef](#)]
2. Dowling, R.; Skabardonis, A.; Carroll, M.; Wang, Z. Methodology for Measuring Recurrent and Nonrecurrent Traffic Congestion. *Transp. Res. Rec.* **2004**, *1867*, 60–68. [[CrossRef](#)]
3. An, S.; Yang, H.; Wang, J. Revealing Recurrent Urban Congestion Evolution Patterns with Taxi Trajectories. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 128.
4. An, S.; Yang, H.; Wang, J.; Cui, N.; Cui, J. Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data. *Inf. Sci.* **2016**, *373*, 515–526. [[CrossRef](#)]
5. Tempelmeier, N.; Feuerhake, U.; Wage, O.; Demidova, E. ST-Discovery: Data-Driven Discovery of Structural Dependencies in Urban Road Networks. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, IL, USA, 5–8 November 2019.
6. Tempelmeier, N.; Sander, A.; Feuerhake, U.; Löhdefink, M.; Demidova, E. TA-Dash: An Interactive Dashboard for Spatial-Temporal Traffic Analytics. In Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3–6 November 2020.
7. Wu, F.; Wang, H.; Li, Z. Interpreting traffic dynamics using ubiquitous urban data. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Francisco, CA, USA, 31 October–3 November 2016.
8. Pan, B.; Demiryurek, U.; Gupta, C.; Shahabi, C. Forecasting Spatiotemporal Impact of Traffic Incidents for Next-generation Navigation Systems. *Knowl. Inf. Syst.* **2015**, 81–88. [[CrossRef](#)]
9. Rettore, P.H.L.; Santos, B.P.; Rigolin, F.; Lopes, R.; Maia, G.; Villas, L.A.; Loureiro, A.A.F. Road Data Enrichment Framework Based on Heterogeneous Data Fusion for ITS. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1751–1766. [[CrossRef](#)]
10. Lee, J.; Hong, B.; Lee, K.; Jang, Y. A Prediction Model of Traffic Congestion Using Weather Data. In Proceedings of the IEEE International Conference on Data Science and Data Intensive Systems, Sydney, Australia, 11–13 December 2015.
11. Chung, Y. Assessment of non-recurrent congestion caused by precipitation using archived weather and traffic flow data. *Transp. Policy* **2012**, *19*, 167–173. [[CrossRef](#)]
12. Liu, Y.; Wu, H. Prediction of Road Traffic Congestion Based on Random Forest. In Proceedings of the 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017.
13. Tempelmeier, N.; Dietze, S.; Demidova, E. Crosstown traffic—supervised prediction of impact of planned special events on urban traffic. *GeoInformatica* **2020**, *24*, 339–370. [[CrossRef](#)]
14. Kwoczek, S.; Martino, S.D.; Nejdil, W. Stuck Around the Stadium? An Approach to Identify Road Segments Affected by Planned Special Events. In Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas de Gran Canaria, Spain, 15–18 September 2015.
15. Anbaroglu, B.; Heydecker, B.; Cheng, T. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. *Transp. Res. Part C Emerg. Technol.* **2014**, *48*, 47–65. [[CrossRef](#)]
16. Laflamme, E.M.; Ossenbruggen, P.J. Effect of time-of-day and day-of-the-week on congestion duration and breakdown: A case study at a bottleneck in Salem, NH. *J. Traffic Transp. Eng. (Engl. Ed.)* **2017**, *4*, 31–40. [[CrossRef](#)]
17. Fouladgar, M.; Parchami, M.; Elmasri, R.; Ghaderi, A. Scalable deep traffic flow neural networks for urban traffic congestion prediction. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017.
18. Gu, Y.; Wang, Y.; Dong, S. Public Traffic Congestion Estimation Using an Artificial Neural Network. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 152. [[CrossRef](#)]
19. Zhu, L.; Krishnan, R.; Guo, F.; Polak, J.W.; Sivakumar, A. Early Identification of Recurrent Congestion in Heterogeneous Urban Traffic. In Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, NZ, USA, 27–30 October 2019.
20. Tseng, F.; Hsueh, J.; Tseng, C.; Yang, Y.; Chao, H.; Chou, L. Congestion Prediction With Big Data for Real-Time Highway Traffic. *IEEE Access* **2018**, *6*, 57311–57323. [[CrossRef](#)]
21. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Short-term traffic forecasting: Where we are and where we are going. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 3–19. [[CrossRef](#)]
22. Xie, D.; Wang, M.; Zhao, X. A Spatiotemporal Apriori Approach to Capture Dynamic Associations of Regional Traffic Congestion. *IEEE Access* **2019**, *8*, 3695–3709. [[CrossRef](#)]

23. Xiong, H.; Vahedian, A.; Zhou, X.; Li, Y.; Luo, J. Predicting Traffic Congestion Propagation Patterns: A Propagation Graph Approach. In Proceedings of the 11th ACM SIGSPATIAL International Workshop on Computational Transportation Science, Seattle, WA, USA, 6 November 2018.
24. Chen, Z.; Yang, Y.; Huang, L.; Wang, E.; Li, D. Discovering Urban Traffic Congestion Propagation Patterns with Taxi Trajectory Data. *IEEE Access* **2018**, *6*, 69481–69491. [[CrossRef](#)]
25. Saeedmanesh, M.; Geroliminis, N. Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks. *Transp. Res. Procedia* **2017**, *23*, 962–979. [[CrossRef](#)]
26. Atluri, G.; Karpatne, A.; Kumar, V. Spatio-Temporal Data Mining: A Survey of Problems and Methods. *ACM Comput. Surv.* **2018**, *51*, 1–41 [[CrossRef](#)]
27. Xu, M.; Wu, J.; Liu, M.; Xiao, Y.; Wang, H.; Hu, D. Discovery of Critical Nodes in Road Networks Through Mining From Vehicle Trajectories. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 583–593. [[CrossRef](#)]
28. Brunauer, R.; Schmitzberger, N.; Rehr, K. Recognizing Spatio-Temporal Traffic Patterns at Intersections Using Self-Organizing Maps. In Proceedings of the 11th ACM SIGSPATIAL Int. Workshop on Computational Transportation Science, Seattle, WA, USA, 6 November 2018.
29. Li, X.; Li, Z.; Han, J.; Lee, J. Temporal Outlier Detection in Vehicle Traffic Data. In Proceedings of the 2009 IEEE 25th International Conference on Data Engineering, Shanghai, China, 29 March–2 April 2009.
30. Asakura, Y.; Kusakabe, T.; Nguyen, L.X.; Ushiki, T. Incident detection methods using probe vehicles with on-board GPS equipment. *Transp. Res. Part Emerg. Technol.* **2017**, *81*, 330–341. [[CrossRef](#)]
31. Taylor, M.A. Critical transport infrastructure in Urban areas: impacts of traffic incidents assessed using accessibility-based network vulnerability analysis. *Growth Chang.* **2008**, *39*, 593–616. [[CrossRef](#)]
32. Scott, D.M.; Novak, D.C.; Aultman-Hall, L.; Guo, F. Network robustness index: A new method for identifying critical links and evaluating the performance of transportation networks. *J. Transp. Geogr.* **2006**, *14*, 215–227. [[CrossRef](#)]
33. Kokoska, S.; Zwillinger, D. *CRC Standard Probability and Statistics Tables and Formulae*; CRC Press: Boca Raton, FL, USA, 2000.
34. Fonseca, L.M.G.; Ii, F.M. Satellite imagery segmentation: a region growing approach. *Simpósio Bras. Sensoriamento Remoto* **1996**, *8*, 677–680.
35. Kuhn, H. The Hungarian Method for the Assignment Problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [[CrossRef](#)]

Short Biography of Authors

Nicolas Tempelmeier is a Ph.D. student at the L3S Research Center in Hannover, Germany. He received his M.Sc. in Computer Science from the Leibniz University Hannover. His research interests lie in the intersection of artificial intelligence and geographic information systems. His research focuses on data-driven analysis of spatial, temporal and Web data.

Udo Feuerhake is a post-doc at the Institute of Cartography and Geoinformatics of the Leibniz Universität Hannover. After studying Computer Aided Engineering, he received his PhD degree for his work on object tracking and movement pattern recognition. His current research interests are in the analysis of spatio-temporal data in a variety of domains, where soccer and transportation stand out.

Oskar Wage is currently a research assistant at the Institute of Cartography and Geoinformatics at the Leibniz University Hannover. Previously, he completed his Bachelor and Master of Science in Geodesy and Geoinformatics there. Already during his studies, he worked on different applications of route guidance and navigation. At the institute, he is doing research in the field of urban transport by analyzing trajectories and other spatial data, especially about alternative concepts for urban logistics and bicycle infrastructure and mobility.

Prof. Dr. Elena Demidova is Professor of Computer Science and head of the Data Science and Intelligent Systems Group (DSIS) at the University of Bonn since October 2020. She received her PhD from the Leibniz University of Hannover in 2013, and her MSc degree in Information Engineering from the University of Osnabrück in a joint program with the University of Twente in 2006. Previously, she worked as Research Group Leader at the L3S Research Center at the Leibniz University of Hannover and as Senior Research Fellow at the University of Southampton. Her main research interests are in Data Science, Machine Learning, the Web and Semantic Web.