applied sciences

MDPI

Article

Urban Congestion Avoidance Methodology Based on Vehicular Traffic Thresholding

Ioan Stan *, Daniel Alexandru Ghere , Paula Iarina Dan and Rodica Potolea *

Department of Computer Science, Technical University of Cluj-Napoca, 26-28 G. Baritiu, 400027 Cluj-Napoca, Romania

* Correspondence: ioan.stan@cs.utcluj.ro (I.S.); rodica.potolea@cs.utcluj.ro (R.P.)

Abstract: Vehicular traffic in urban areas faces congestion challenges that negatively impact our lives. The infrastructure associated with intelligent transportation systems provides means for addressing the associated challenges in urban areas. This study proposes an effective and scalable vehicular traffic congestion avoidance methodology. It introduces a traffic thresholding mechanism to predict and avoid vehicular traffic congestion during route computation. Our methodology was evaluated and validated by employing four road network topologies, three vehicular traffic density levels and various traffic light configurations, resulting in 26 urban traffic scenarios. Using our approach, the number of vehicles that can run in free flow can be increased by up to 200%, whereas for traffic congestion scenarios, the time spent in traffic may be reduced by up to 69% and CO_2 emissions by up to 61%. To the best of our knowledge, in the vehicular traffic flow prediction domain, this is the first approach that covers a set of road network topologies and a large and representative set of scenarios for simulated urban traffic congestion testing. Moreover, the comparative analysis with different other solutions in the domain, showed that we obtained the best driving time and CO_2 emission reduction.

Keywords: road network topology; traffic prediction; route computation; congestion avoidance; traffic simulation

check for updates

Citation: Stan, I.; Ghere, D.A.; Dan P.I.; Potolea, R. Urban Congestion Avoidance Methodology Based on Vehicular Traffic Thresholding. *Appl. Sci.* **2023**, *13*, 2143. https://doi.org/10.3390/app13042143

Academic Editors: Roland Jachimowski and Andrzej Ratkiewicz

Received: 5 January 2023
Revised: 2 February 2023
Accepted: 3 February 2023
Published: 7 February 2023

Copyright: © 2023 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

Nowadays, most of the urban areas face traffic congestion, which decreases the living standard and increases the time spent in traffic by drivers: extra time sitting in traffic (in terms of billions of hours), resulting in hundreds of billions of dollars costs [1,2]. For example, as stated by the authors in [3], in major urban areas in the United States (US), the travel time during congestion is increased by 32%. In recent years, intelligent transportation systems (ITS) have attempted to provide solutions for traffic prediction and vehicular routes enhancements for the purpose of congestion avoidance. Many of these solutions are based on algorithms that use knowledge from a large spectrum of data. The large volume of data used to improve traffic in urban areas faces another problem, scalability, as is shown in [4]. Data sharing between vehicles affects scalability. More specific data sharing studies based on various architectures and communication methodologies are presented in [5–7]. In [8,9], multiple scalable data models were used to represent vehicular data for traffic prediction and congestion avoidance. These approaches use a centralized architecture that stores and controls vehicular traffic information in order to globally reduce traffic congestion in urban areas. The prediction of ITS' vehicular traffic and the avoidance of vehicular traffic congestion are affected by the route computation process. The route computation process uses and affects vehicular traffic aspects that are reflected in the daily travel experience, as has been shown in studies such as [10–13]. From an implementation cost perspective, traffic simulation is an efficient and reliable platform that has been used to model and run vehicular traffic in many studies [8,9,14–20].

The work in [8,9] modeled vehicular traffic in a scalable manner, but the congestion avoidance improvement was below 3%. This is because the route computation algorithm

uses a simple penalization model for traffic congestion. Moreover, the existing congestion avoidance solutions lack variety regarding the tested urban road network topologies, similar to the ones presented in [21]. In addition, in the existing solutions, most of the cases simulated low to moderate congestion but did not simulate high congestion.

To summarize, all of the aforementioned studies lack at least one of the following properties:

- Effectiveness of the solution regarding vehicular traffic congestion avoidance and traffic flow improvement;
- Scalability regarding the amount of vehicular traffic data representation and simulation;
- Comprehensiveness of the evaluated vehicular traffic scenarios.

Considering the aforementioned achievements and challenges, we considered that it is valuable to design and develop a novel vehicular traffic congestion avoidance solution that noticeably improves moderate and high traffic congestion simulated on a set of comprehensive road network topologies.

The main goal of this study is to develop a congestion avoidance solution based on a vehicular traffic thresholding mechanism that improves urban traffic congestion in a comprehensive set of simulated traffic scenarios. The main contributions of this study are as follows:

1. A novel vehicular traffic congestion avoidance approach was used in a Vehicle to Cloud (V2C) simulated ecosystem;
2. A novel traffic thresholding mechanism was integrated into the route computation algorithm in order to compute routes that avoid traffic congestion generation. The routes computation was carried out on a comprehensive set of road network topologies representing real urban areas. For evaluation, we simulated vehicular traffic in SUMO based on the computed routes;
3. Defining and evaluating a set of testing scenarios for tuples: <vehicular density, road network topology, infrastructure settings>.

The remainder of this paper is organized as follows: The next section discusses work related to the existing literature. In Section 3, we present a novel vehicular traffic congestion avoidance methodology that contains the proposed vehicular traffic thresholding mechanism and its corresponding algorithms. The Section 4 presents the SUMO-based simulation framework used to implement the proposed methodology for congestion avoidance, the evaluation of the proposed methodology and discussion of the experimental results. In the last section, we conclude the study and present an overview for further work.

2. Related Work

Vehicular traffic and traffic congestion have been discussed, analysed, and evaluated in various studies in previous years. First, we discuss the related work in the literature regarding prediction of vehicular traffic. Second, we move our discussion more specifically to focus on large-scale congestion simulation in urban areas.

2.1. Vehicular Traffic Prediction

An important step in vehicular congestion avoidance is the prediction of traffic. Many vehicular traffic prediction approaches from the literature use data mining and simulation for the purpose of predicting short-term vehicular traffic prediction in non-urban areas' contexts. A comprehensive review in this matter was performed in [8]. The work in [22] proposed a solution to predict the fundamental traffic diagram parameters [23,24]. Urban traffic prediction solutions are fewer than for non-urban ones. The solution in [25] proposed a solution to predict travel time by using a large amount of traffic information from Berlin and Thessaloniki.

In addition to traffic prediction approaches, there are more specific solutions used to predict vehicular traffic congestion in order to reduce the time spent in vehicles. In this

regard, the authors in [26] propose an unsupervised incremental learning solution that can be used to detect and profile vehicular traffic congestion. They predicted short-term traffic by using hundreds of millions of vehicular movements obtained from intersections in the State of Victoria from Australia.

A promising solution for evaluating congestion was proposed in [27]. Their strategy is based on geocoding vehicular traffic events that come from Twitter in order to obtain a training data set. The training data set is to obtain a prediction model by applying a support vector machine method. This approach was used to generate a spatio-temporal vehicular traffic data set in order to analyse vehicular traffic congestion in Mexico City. The solution proposed in [28] is based on various re-routing mechanisms used to avoid vehicular traffic congestion. For their experiments, they simulated one thousand vehicles by using SUMO [29] and TraCI [30]. In [31,32], an inter-vehicular communication based solution is proposed to avoid vehicular traffic congestion in urban areas. For the work in [31], the New York map was used for traffic simulations and measurements, while for the one in [32], the evaluation was performed in Colima city from Mexico by combining inter-vehicular communication, infrastructure-to-infrastructure connectivity, fixed roadside infrastructure and a large amount of data. The authors in [33] propose a vehicular traffic congestion avoidance method based on vehicle re-routing in the context of synthetic grid maps. Their method wants to provide an optimal vehicular guidance suggestion to bypass congested roads as soon as possible.

The impact of humans on traffic flow was evaluated in [34], and it was concluded that traffic stability is significantly impacted by humans and, therefore, the risk of traffic breakdowns is imminent.

The study in [35] proposes a hybrid deep learning model to predict vehicular traffic flow. In order to automatically determine the importance of input parameters of large scale space-time points, they introduce an attention model aimed to measure the importance of input traffic flow of past spatial-temporal position in correlation with the future traffic flow. A noteworthy vehicular traffic modelling approach is described in [36]. Their approach models and analyses vehicular traffic at a specific signalized road intersection by using an artificial neural network (ANN) model based on traffic flow parameters such as vehicular speed, time of the day, traffic volume, and class of vehicles.

A special category of traffic congestion is the one that is generated by traffic incidents. The studies in [37–40] propose approaches that can be used to improve the urban traffic congestion prevention, especially in the context of signalized intersections.

In addition to comprehensive work about traffic simulation benefits and challenges in the context of TraffSim [41,42], a noteworthy study on vehicular traffic prediction and congestion reduction through route computation, was presented in [43]. It simulates between 2000 and 3000 vehicles on real and synthetic maps. Their approach resembles the study in [8] and our proposal, particularly when using a route computation algorithm to predict and avoid congestion. The study in [44] analyses and presents the benefits of cooperative route computation in comparison with individual (egoistic) route computation. Cooperative route computation is also one of the main pillars of the work in [8] and this study.

2.2. Large Scale Urban Traffic Congestion Simulation

The discussion and analysis in the previous subsection were about prediction of vehicular traffic and avoidance of vehicular traffic congestion, especially in urban areas. In this subsection, the large-scale urban traffic congestion simulation approaches proposed by different authors are discussed. To model real-world traffic congestion scenarios in urban areas through simulation, we need to take into consideration the scalability of the model. A summary of the existing studies on urban traffic congestion simulation is presented in Table 1. For simplicity, in the aforementioned table and in this study, we use a “K” symbol when we refer to thousand.

Table 1. Large scale urban traffic congestion simulation solutions.

Solution	Simulator	Road Network	Area Size	Number of Simulated Vehicles	Maximum of Concurrent Vehicles *
VNS [15]	DIVERT	Porto City	N/A	130 K	15 K
Elbery [16,17]	INTEGRATION	Los Angeles	133 km^2	563 K+	30 K
Farag [18]	INTEGRATION	Los Angeles	133 km^2	145 K	30 K
FOXS [45]	SUMO	Ottawa, Cologne	8 km^2 and 4 km^2	2.2 K and 14 K	N/A
SOPHIA [46]	SUMO	Cologne	approx. 25 km^2	46 K	N/A
Re-RouTE [47]	SUMO	Los Angeles, Paris	25 km^2	1.25 K to 6.25 K	N/A
ATOM [48]	SUMO	Baltimore	approx. 1 km^2	7.2 K	N/A
Stan [8,19]	OsmAnd	New York, Cluj-Napoca	N/A	10 K and 20 K	N/A
CAVTTM	SUMO	Barcelona, Bucharest, New York, Tokyo	65 km^2	50 K+ and 100 K+	49 K

* Peak number of concurrent vehicles running on the roads simultaneously. N/A: not applicable.

The vehicular network simulator (VNS), described in [15], uses a DIVERT 2.0 traffic simulator to represent traffic that simulates a total of 130 K vehicles, having a maximum of 15 K concurrent vehicles running on the road network. Their methodology is based on quad tree data structure that is used to store and represent a large number of vehicles on the Porto City road network.

The authors of study in [16,17] use the INTERGRATION to accurately simulate large-scale traffic on the road network of downtown Los Angeles. A total number of vehicles that run in their solution is more than 563 K, with a maximum of 30 K concurrent vehicles on the road network. Considering that the road network area used for the simulation was 133 km^2, they managed to simulate moderate congestion on the roads. The authors of [18] also use an INTEGRATION traffic simulator to model large-scale urban traffic. The total number of vehicles that run in their solution is more than 145 K, having 30 K concurrent vehicles on the road network in downtown Los Angeles. Because the road network has an area of 133 km^2, the simulated traffic mimics a moderate congestion scenario. Their vehicle simulation mechanism performed better than the approaches in [16,17] because the vehicle positions were modelled using a grid cell data structure with an update index. In this case, the query operation has a linear performance with regard to the vehicular density, whereas we have a constant complexity for update operation. A bottleneck in this solution is represented by the query operation when the frequency of query requests is high, or the vehicular density is high (as in the vehicular congestion case).

The work in [45] simulates traffic congestion with 2.2 K vehicles on a road network of 8 km^2 in Ottawa and 14 K vehicles on a road network of 4 km^2 in Cologne. The maximum number of concurrent vehicles on the road network was not specified. Another work that simulates traffic congestion in Cologne was described in [46] and simulated 46 K vehicles on a road network of 25 km^2 in Cologne. As in [45], they did not specify the maximum count of concurrent vehicles on the roads. The work in [47] generated low level congestion (between 1.25 and 6.25 K vehicles) on two road networks of 25 km^2 in Los Angeles and Paris. The Paris area is more prone to traffic congestion owing to the presence of more bridges. On the other hand, in [48], 7.2 K vehicles in 1 km^2 to generate high congestion are simulated. The works in [45–47] presented their percentage improvement regarding the time spent in traffic, as shown in Section 5.

More specific traffic modelling approaches are described in [8,19]. In these cases, the traffic state is defined during vehicle route computation and stored using range query data structures. The vehicle simulation was performed by adapting the OsmAnd navigation application to compute vehicular routes by considering the current state of the traffic. The current state of the traffic is obtained by querying the vehicular density on a road segment at a specific moment. The range query data structure used for storage, query and update of vehicular traffic information is segment tree for the work in [19] and K-ary tree for the solution in [8]. The segment tree lacks scalability due to query and update operations performance, the K-ary tree, especially the K-ary entry point (KEP) tree, scales efficiently and without any bottlenecks due to high performing query and update operations.

As is shown in [8], the KEP tree data structure performs better than any other scalable traffic modelling solutions. The number of simulated vehicles in [19] is 10 K, while the number of simulated vehicles in [8] is 20 K. The road network size used for the simulation and the maximum number of concurrent vehicles are not specified in [19] nor in [8].

3. Vehicular Traffic Congestion Avoidance Methodology

Congestion avoidance based on the vehicular traffic thresholding mechanism (CAVTTM) integrates in SUMO the KEP tree data structure that we developed in [8] and proposes a novel vehicular traffic thresholding mechanism in order to reduce or avoid congestion in urban areas. To evaluate the efficiency of CAVTTM, we generated and evaluated a number of relevant urban vehicular traffic scenarios by combining the parameters below:

- Vehicular density represented by the number of simulated vehicles;
- Road network topology classified in four types: Grid Topology, Historical/Irregular Topology, Unbalanced Grid Topology, Hybrid Topology;
- Infrastructure settings represented by the average traffic light delay, which represents the average time spent at each traffic light (seconds).

Thus, based on the above parameters, we define the tuple <vehicular density, road network topology, infrastructure settings> used by our methodology. The defined tuple is employed by the proposed methodology, as shown in Figure 1. The route computation algorithm uses data stored statically in the map (e.g., segment length, speed limit, and vehicular infrastructure, number of lanes) and infrastructure information (e.g., traffic light delay) to compute routes. The road network topology is based on the urban area used to compute routes and influences the output of the route computation algorithm. We define four types of road network topologies as follows:

- *Grid Topology* representing almost ideal urban areas with dense road topology in the form of a grid, surrounded by highways and a proportional surface;
- *Historical/Irregular Topology* representing urban areas with a small density of road infrastructure following random geometry based on historical roads and proportional surface;
- *Unbalanced Grid Topology* representing urban areas with long roads, highways, grid topology, disproportional surface and pedestrian areas;
- *Hybrid Topology* representing urban areas with very rich road infrastructure that has many highways, large intersections and a proportional surface.

The route computation algorithm outputs the computed routes which, at their turn, generates the vehicular traffic density to be considered during future route computation. For the vehicular density, we define *free flow* for a range of [10 K+ ...30 K+] simulated vehicles and *congestion* for a range of [50 K+ ...100 K+] simulated vehicles. Both free flow and congestion are generated on urban areas of 65 km ^{2}.

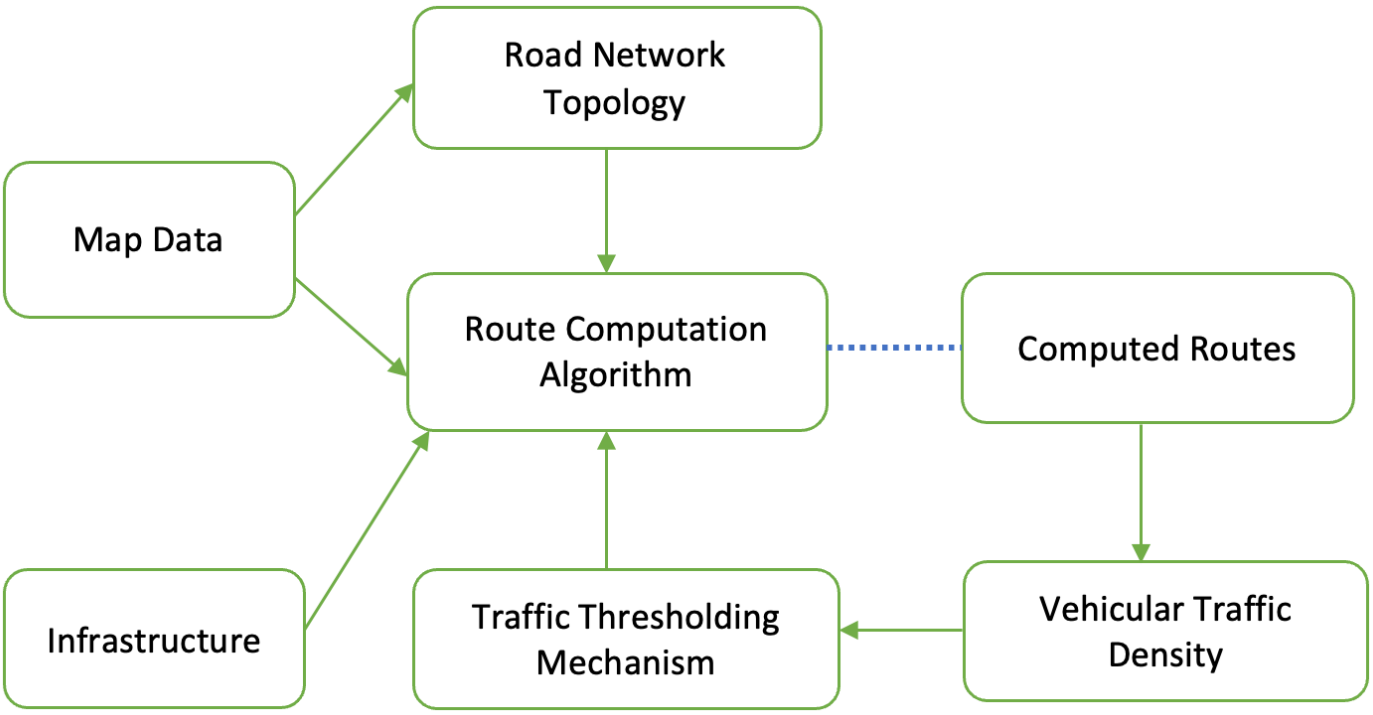


Figure 1. Flowchart of the main concepts employed in the methodology.

In is noteworthy that, in addition to the comprehensiveness of the road network topologies, our focus is to reach a very high congestion by having up to 49 K concurrent vehicles that run on a road network. To the best of our knowledge, this is the highest number of concurrent vehicles simulated in a road network in order to model traffic congestion.

The traffic thresholding mechanism is a novel concept introduced in this study and is employed in the route computation algorithm to predict possible traffic congestion and to propose alternative routes in order to avoid congestion as much as possible. In case of possible traffic congestion, the thresholding mechanism increases the effort on a segment and, therefore, the route computation algorithm searches for other lower effort segments in order to balance the vehicular traffic.

The congestion avoidance solution we propose is described in the flow diagram shown in Figure 2.

To provide an effective, scalable and reliable solution for congestion avoidance, we designed a layered flow based on a set of successive steps defined below:

- S1—Generation of the start and destination points for each vehicular route in the context of a selected road network topology from a comprehensive set of road network topologies;
- S2—The output of S1 is consumed by the route computation algorithm to obtain routes that reduce or avoid vehicular traffic congestion. During route computation, the proposed thresholding mechanism targeting vehicular traffic congestion avoidance is employed. In addition, for vehicle position estimation in a scalable manner, we use the KEP Tree data structure that we proposed in [8];
- S3—The computed routes in S3 were used to run vehicles on the selected road network topology in a vehicular traffic simulation environment. The vehicular traffic simulation environment was adapted to log vehicular traffic data that is parsed in the next step;
- S4—The generated vehicular traffic logs generated in S3 are parsed to provide data for evaluating the proposed solution in terms of congestion reduction and avoidance.

The route computation algorithm presented in Algorithm 1 takes the start and destination points inputs and finds the context-specific best route, adapting Dijkstra's algorithm by integrating the KEP tree and the proposed traffic thresholding mechanism. Beginning with

the starting point, it uses a priority queue to extend the route search graph by considering the unvisited neighbours of the first element from the queue. It selects the best match for the next road segment, which may be part of the route. The best match represents the road segment with the minimum effort value. To avoid traffic congestion, we defined a novel thresholding mechanism that penalizes efficiently the computed effort on a road segment to avoid congestion when possible. To achieve this, it is necessary to know the traffic state of each segment at any moment. For this purpose, we use the KEP tree for storage, and query and update the vehicular traffic state in a simulated ecosystem. The KEP tree data structure used to model vehicular traffic scales very well and is easy to implement. The algorithms that correspond to the query and update operations are described in [8].

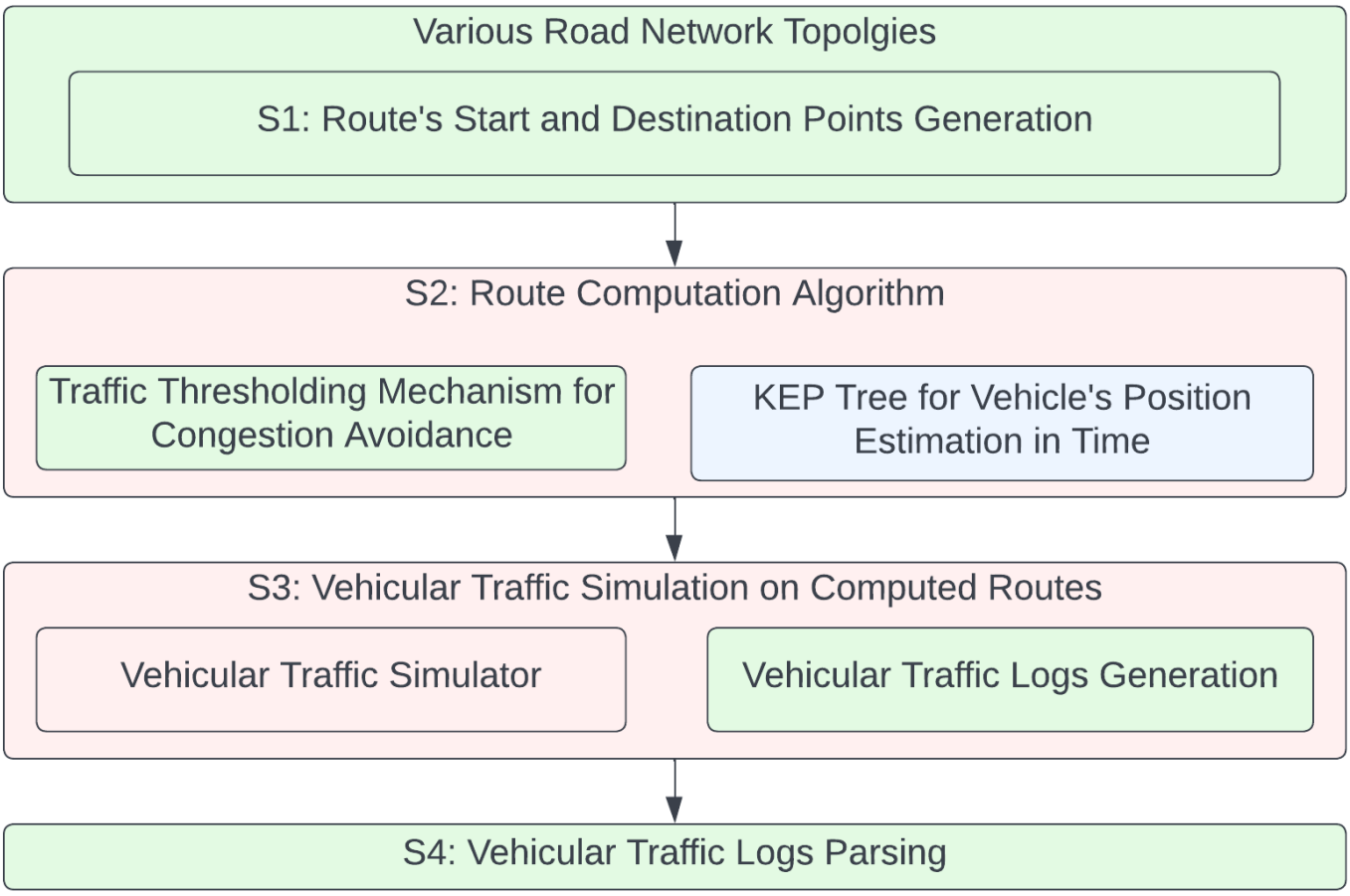


Figure 2. Flow diagram of vehicular traffic congestion avoidance solution.

The following notations and concepts are used as a model by Algorithm 1 and its sub-routines that use the traffic thresholding mechanism to avoid congestion:

- p_s—starting point of a route given as GPS coordinates;
- p_d—destination point of a route given as GPS coordinates;
- $segment_s$ and $segment_d$—road segments corresponding to p_s and p_d, respectively. The road segments are found through map matching of p_s and p_d, respectively. Each segment is represented by two GPS points that correspond to the begin and the end of the segment;
- t_0—variable representing the timestamp when a specific vehicle starts to run on its planned route;
- t_s—variable representing the timestamp when a specific vehicle starts to run on any segment. Used as a local variable in the algorithms;

- t_e—variable representing the timestamp when a specific vehicle leaves any segment. Used as a local variable in the algorithms;
- ETT—variable representing the current estimated travel time until passing any segment. Used as a local variable in the algorithms;
- route(p_s, p_d)—computed route between p_s and p_d represented by a set of consecutive connected road segments;
- δ—variable representing the average time spent for each traffic light (seconds). This has a specific value for each urban area;
- $E_{segment}$—variable representing the already computed effort value corresponding to a segment in a computed route. Its value is computed during route computation algorithm;
- $N(node)$—set of not visited neighbours of node. Each neighbour is represented by a road segment;
- E_{max}—numeric constant value representing the maximum effort factor used during effort computation (used to reduce graph exploration);
- S_{min}—numeric constant value representing the minimum speed factor used for speed computation in the case of congestion;
- D_{max}—numeric constant value representing the maximum delay factor used for delay computation in the case of congestion;
- $L_{vehicle}$—numeric constant value representing the average length of the space needed by a vehicle on the road. In this paper, we considered 7 m;
- $L_{segment}$—variable representing the length of a specific road segment in meters;
- $lanesCount_{segment}$—variable representing the number of lanes on a road segment. Its value depends on the corresponding road segment;
- t—variable representing the estimated timestamp when a vehicles enters on a road segment. Used as a local variable in the algorithms;
- $vehiclesCount_{segment}(t)$—variable representing the number of vehicles on a road segment at a specific moment t. It is used as a local variable in the algorithms;
- $\rho_{segment}(t)$—variable representing the predicted vehicular density on a road segment at a timestamp, defined as

$$\frac{vehiclesCount_{segment}(t) \times L_{vehicle}}{L_{segment} \times lanesCount_{segment}} \quad (1)$$

  and vehicular density can have value in interval $[0 \dots 1]$. Used as a local variable in the algorithms;
- Threshold levels represented by a set of constant numerical values—for the purpose of providing vehicular routes that avoid congestion, we use ten density threshold levels defined by the set. We found through testing that it is necessary and sufficient to split the density in ten threshold levels in order to obtain significant improvement on congestion reduction and avoidance:

$$T = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}; \quad (2)$$

- F_e—the set of numeric values representing the effort factors used to penalize congested road segments based on vehicular traffic density threshold levels in order to route traffic on less used road segments. It was experimentally identified by trying values between 1 and 100 and analysing the evolution of the traffic in time;
- $F_e(\rho)$—effort factor function based on vehicular traffic density threshold levels. It has 10 values;
- $E_{segment}(\rho)$—the effort of a segment to become part of a route (during route computation algorithm), defined as a function based on vehicular traffic density threshold levels

$$(E_{segment} + \delta) \times F_e(\rho) \quad (3)$$

- $\text{speedLimit}_{\text{segment}}$—variable representing the speed limit for a specific segment from the road network. Its value depends on the corresponding road segment;
- F_s—the set of numeric values representing the speed factors used to compute the speed of a vehicle based on vehicular traffic density threshold levels. This was experimentally identified by following the fundamental traffic flow speed-density diagram [23,24];
- $F_s(\rho)$—speed factor function based on vehicular traffic density threshold levels. It has 10 values;
- $S_{\text{segment}}(\rho)$—computed speed of a segment defined as a function based on vehicular traffic density threshold levels

$$\text{speedLimit}_{\text{segment}} \times F_s(\rho); \tag{4}$$

- F_d—the set of numeric values representing the delay factors used to compute the vehicular traffic delay based on vehicular traffic density threshold levels and δ. This was experimentally identified by comparing the estimated travel time with the corresponding metric of existing navigation solutions (e.g., Google Maps).
- $F_d(\rho)$—delay factor function based on vehicular traffic density threshold levels. It has 10 values;
- $D_{\text{segment}}(\rho)$—the delay introduced by the vehicular traffic on a specific segment (in seconds), defined as a function based on vehicular traffic density threshold levels

$$\delta \times F_d(\rho). \tag{5}$$

The route computation algorithm integrates the estimates of vehicular density, travel effort, speed and traffic delay defined above by the formulas (1)–(5). The following part of this section presents their corresponding algorithms. The vehicular density computation is performed in Algorithm 2 and uses the KEP tree query operation to count vehicular density on a segment at a specific moment. Details on KEP Tree data structure and its associated operations (store, update, query) may be found in [8]. The KEP tree scales appropriately and fits the specific congestion avoidance needs. Based on this, in this work, we propose a novel traffic thresholding mechanism that is used to model the reality and efficiently balance the traffic on a specific road network. The thresholding mechanism is based on the threshold levels defined above, which are used during effort, delay and speed computation in the context of the route computation algorithm. In this way, possible congested road segments are penalized, and traffic is routed on less used road segments. The vehicular density of a segment is computed in Algorithm 2.

The effort used during segment evaluation in Algorithm 1 is computed in Algorithm 3 and is based on vehicular traffic threshold levels defined above. For each threshold value, we associate a corresponding effort factor value that is used to penalize congested road segments and to route traffic on less used road segments.

Algorithm 4 computes the speed of a vehicle by considering the speed limits and δ in combination with the threshold levels and speed factors. For each threshold value, we associated the corresponding speed factor.

Algorithm 5 computes the vehicular traffic delay by considering δ in combination with threshold levels and delay factors. For each threshold value, we associated the corresponding delay factor. Vehicular traffic delay is used to estimate the time interval during which a simulated vehicle is on a specific road segment.

The computational complexity of the Algorithms 3–5 is linear because they are searching linearly the first density threshold value that is greater than a specific value δ. Theoretically, this can be achieved by using binary search that is having a logarithmic time. Still, we found in practice that, because we have only 10 threshold values, the complexity constant of binary search algorithm does not add any gain to our implementation.

Algorithm 1 Route computation algorithm

procedure COMPUTE ROUTE (p_s, p_d)
  $init(queue)$
  $segment_s \leftarrow getSegment\ (p_s)$
  $segment_d \leftarrow getSegment\ (p_d)$
  $node \leftarrow segment_s$
  $queue.push\ (node)$
  $t_s \leftarrow t_0$
  $ETT \leftarrow t_0$
  while queue is not empty **do**
    $node \leftarrow queue.pop\ ()$
    if $node == segment_d$ **then**
      for each segment in route **do** (p_s, p_d)
        $speed \leftarrow ComputeSpeed\ (t_s, segment)$
        $d \leftarrow ComputeDelay\ (t_s, \delta, segment)$
        $t_e \leftarrow t_s + L_{segment} / speed + d$
        $segment.Update\ (t_s, t_e)$
        $t_s \leftarrow t_e$
      end for
    end if
    $ETT \leftarrow getTravelTime\ (node)$
    for each segment in N (node) **do**
      $effort \leftarrow computeEffort\ (ETT, segment)$
      if $effort < E_{segment}$ **then**
        $E_{segment} \leftarrow effort$
      end if
    end for
  end while
end procedure

Algorithm 2 Compute density

function COMPUTEDENSITY (t, *segment*)
  $vehiclesCount_{segment} \leftarrow segment.Query\ (t)$
  return $(\text{vehiclesCount}_{\text{segment}} \times \text{L}_{\text{vehicle}}) /$
  $(\text{L}_{\text{segment}} \times \text{lanesCount}_{\text{segment}})$
end function

Algorithm 3 Compute Effort

function COMPUTE EFFORT (t, *segment*)
  $\rho \leftarrow ComputeDensity\ (t, segment)$
  $effort \leftarrow E_{segment} + \delta$
  $F_e \leftarrow \{1, 5, 10, 15, 20, 25, 30, 35, 40\}$
  $E_{max} \leftarrow 50$
  for each i in T.size () **do**
    if $\rho < T[i]$ **then**
      return $effort \times F_e[i]$
    end if
  end for
  return $effort \times E_{max}$
end function

Algorithm 4 Compute Speed

```
function ComputeSpeed (t, segment)
    ρ ← ComputeDensity (t, segment)
    F_s ← {1, 0.97, 0.87, 0.64, 0.42, 0.25, 0.17, 0.11, 0.07}
    S_min ← 0.02
    for each i in T.size () do
        if ρ < T[i] then
            return speedLimit_segment × F_s[i]
        end if
    end for
    return speedLimit_segment × S_min
end function
```

Algorithm 5 Compute delay

```
function ComputeDelay (t, δ, segment)
    ρ ← ComputeDensity (t, segment)
    F_d ← {1, 2, 3, 4, 5, 6, 7, 8, 9}
    D_max ← 10
    for each i in T.size () do
        if ρ < T[i] then
            return δ ∗ F_d[i]
        end if
    end for
    return δ ∗ D_max
end function
```

4. Urban Traffic Scenarios' Evaluation and Discussion

4.1. Implementation Framework

The framework used to implement the CAVTTM solution is shown in Figure 3 and reflects the steps of the flow that correspond to the vehicular traffic congestion avoidance solution presented in Figure 2, as follows:

- S1 is implemented using a stand-alone routine that generates GPS positions for the start and destination points on a real urban map data region from Open Street Map (OSM), which fulfills the proposed road network topology requirements;
- For the implementation of S2, we adapted the existing route computation algorithm from SUMO to use the proposed traffic thresholding mechanism and KEP tree data structure in order to reduce or avoid congestion;
- The S3 step implementation is mostly achieved using SUMO components, and for the computed routes, we run vehicle simulations on the selected road network region from OSM using the SUMO simulator. Specific vehicular traffic data obtained during the simulation were logged by using a customized SUMO traffic logger. We adapted a customized traffic logger from SUMO to fulfill our requirements;
- Finally, the generated logs are parsed in order to obtain data for the CAVTTM evaluation in terms of congestion reduction and avoidance.

SUMO was preferred over other vehicle simulation tools (e.g., INTEGRATION and OsmAnd) because of the following main aspects:

- Scalability: SUMO supports hundreds of thousands of simulated vehicles, similar to INTEGRATION, while OsmAnd supports only tens of thousands;
- Simulation time: SUMO is ten times faster than OsmAnd;
- Real maps: SUMO supports real maps (such as INTEGRATION and OsmAnd);
- Research employment: SUMO is the most frequent simulator in the field.

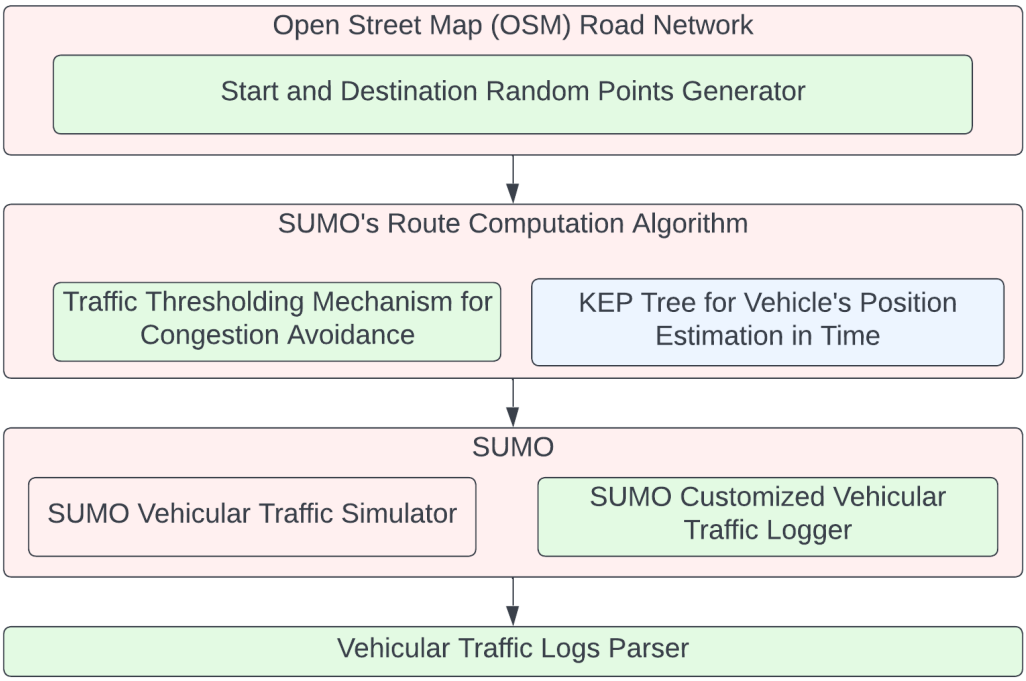


Figure 3. Vehicular traffic congestion avoidance implementation framework.

To implement the vehicular traffic congestion avoidance solution, we injected a few subroutines in SUMO's route computation algorithm in order to compute the density, effort, speed and delay. The injected subroutines computation uses a series of parameters from the proposed thresholding mechanism. Another important aspect regarding the SUMO simulator is the fact that, in order to model vehicular traffic reality, it was necessary to add an average delay (δ) value for each traffic light that is passed by a specific simulated vehicle. An efficient range of δ values was determined by testing the topology of each road network.

4.2. Evaluation Metrics and Scenarios

To address efficiency, usability and performance of the vehicular traffic thresholding mechanism used to avoid congestion, the following metrics were evaluated and discussed:

- s_{av}—the average vehicle speed (km/h);
- d_{total}—the total distance travelled by all the running vehicles in one simulated scenario (km);
- t_{total}—the total travelled time of all the running vehicles in one simulated scenario (hours);
- $t_{simulation}$—the simulation time for all simulated vehicles in one simulated scenario (hours);
- $fuel_{total}$—the total fuel consumption for all vehicles in one simulated scenario (l);
- $CO_{2_{total}}$—the total CO_2 emissions for all vehicles in one simulated scenario (kg);
- $countV_{peak}$—the peak number of concurrent vehicles running on roads in one simulated scenario.

The running environment was a Windows 10 computer with a Ryzen 9 5900X 12-core processor and 32 GB of RAM and a Samsung EVO 980 PRO 2T SSD. The SUMO based simulation framework presented in the previous section runs on this machine in order to simulate vehicular traffic (especially congestion) that runs on a comprehensive set of urban road network topologies. For each simulated vehicle, the start and destination points of the corresponding route were randomly generated. The generated start and destination points belong to the selected urban area.

For simplicity, we use the following notations in the rest of the article:

- SUMO—default SUMO solution used to simulate vehicles;

- CASUMO$_\delta$—our congestion avoidance solution integrated with default SUMO and considering that a vehicle is delayed on average by δ seconds at each traffic light. It applies Algorithm 1 with its subroutines to a specific road network topology to generate vehicular routes that avoid congestion. During experiments, for each network topology, three δ values were used.

The evaluation of our thresholding mechanism for vehicular traffic congestion avoidance is defined based on scenarios driven by parameters corresponding to the above defined tuples:

- Number of Vehicles—representing the total number of vehicles during a given simulation:
  - Free flow traffic that simulates vehicles in the range of [10 K+...30 K+];
  - More than 50,000 (50 K+) vehicles during 2.8 h for moderate to high traffic congestion;
  - More than 100,000 (100 K+) vehicles during 2.8 h for very high traffic congestion.
- Road Network Topology—we simulated vehicles using different urban areas in order to cover the road network topologies presented in [21]. These are the orange zones shown in Figure 4:
  - Grid topology: Barcelona—almost ideal urban area with dense road topology in the form of a grid, surrounded by highways and a proportional surface;
  - Historical/Irregular Topology: Bucharest—urban areas with a small density of road infrastructure following random geometry based on historical roads and proportional surface;
  - Unbalanced Grid Topology: New York—urban area with long roads, highways, grid topology, disproportional surface and pedestrian subareas;
  - Hybrid Topology: Tokyo—urban area with very rich road infrastructure that has many highways, large intersections and a proportional surface.
- δ—average time (in seconds) spent by each vehicle at a traffic light during simulation. This is a natural number that is in the range of [1 ... 9] seconds and was experimentally identified for each road network topology.

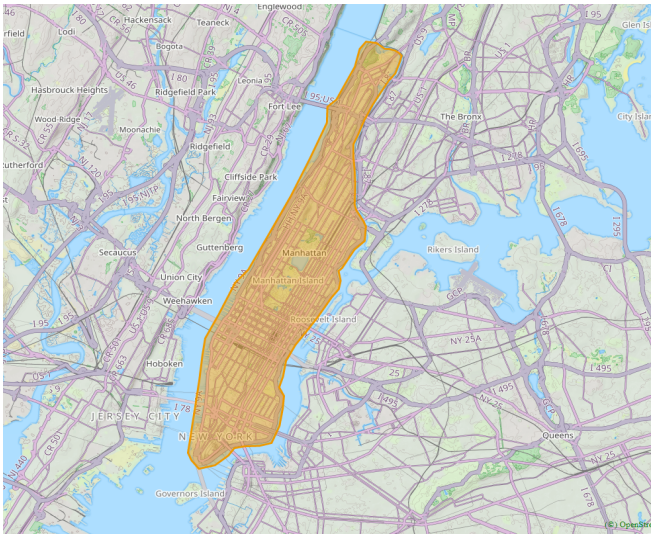
4.3. Scenarios with 50 K+ Simulated Vehicles

For vehicular scenarios with 50 K+ simulated vehicles (moderate vehicular traffic congestion), we run one test with SUMO and three tests with δ for each urban area. The δ values were selected to reduce t_{total} as much as possible. Two other important metrics that evolve with t_{total} are $fuel_{total}$ and CO_2 emissions ($CO_{2_{total}}$). It is interesting to observe the s_{av} evolution that depends on t_{total} and d_{total} because, in most of the cases, d_{total} is inversely with t_{total} and, therefore, s_{av} improvement is very high. For each scenario that runs 50 K+ vehicles in the selected urban regions, we present the following:

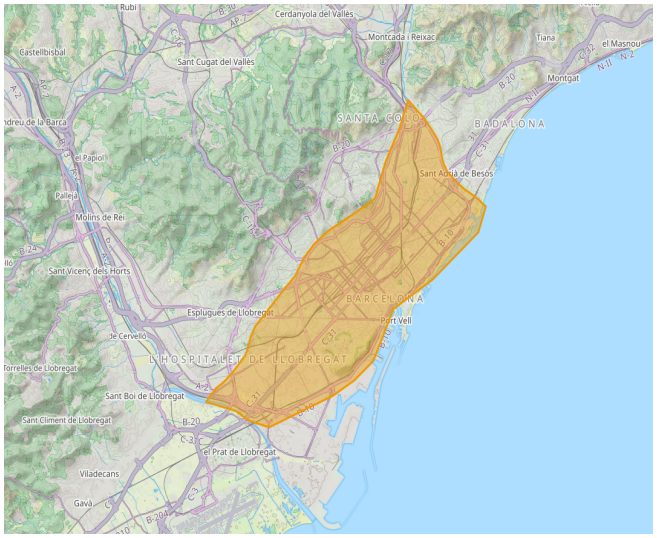
- A table with absolute values obtained for the above defined metrics;
- A table with improvement values (in percentage) obtained by our solution in comparison with SUMO;
- A chart with speed distribution in clusters is presented for the δ value that corresponds to the minimum value of t_{total}.

4.3.1. Grid Topology—Barcelona

As can be observed from Tables 2 and 3, for moderate congestion in Barcelona, we obtained more than a 37% improvement for the time spent in traffic, while the improvement of average speed was more than 77%. Figure 5 shows that the traffic has a good flow because most of the vehicles' speeds are above 10 km/h.



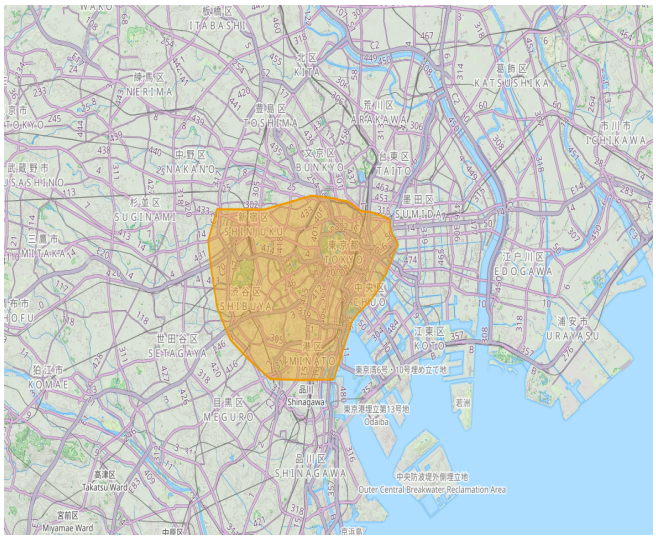
(**a**) NewYork



(**b**) Barcelona



(**c**) Bucharest



(**d**) Tokyo

Figure 4. Maps of simulation areas. Orange shaded the highly congested area (65 km^2).

Table 2. Results for 50 K+ simulated vehicles in Barcelona.

Metric	SUMO	CASUMO$_1$	CASUMO$_2$	CASUMO$_3$
s_{av}	10.90	19.29	19.78	19.94
d_{total}	320,524.76	353,702.60	372,673.01	384,579.08
t_{total}	29,415.87	18,333.38	18,841.57	19,291.14
$t_{simulation}$	6.29	4.91	4.93	4.76
$fuel_{total}$	97,945.29	63,614.5	65,148.6	66,319.91
$CO2_{total}$	227,660.31	147,909.91	151,479.26	154,204.95
$countV_{peak}$	11,560	7242	7494	7693

Table 3. Percentage Improvement of the proposed solution for 50 K+ simulated vehicles in Barcelona.

Metric	$CASUMO_1$	$CASUMO_2$	$CASUMO_3$
s_{av}	77.06	81.52	82.96
d_{total}	−10.35	−16.27	−19.98
t_{total}	37.68	35.95	34.42
$t_{simulation}$	21.93	21.55	24.37
$fuel_{total}$	35.05	33.48	32.29
$CO_{2_{total}}$	35.03	33.46	32.27
$countV_{peak}$	37.35	35.17	33.45

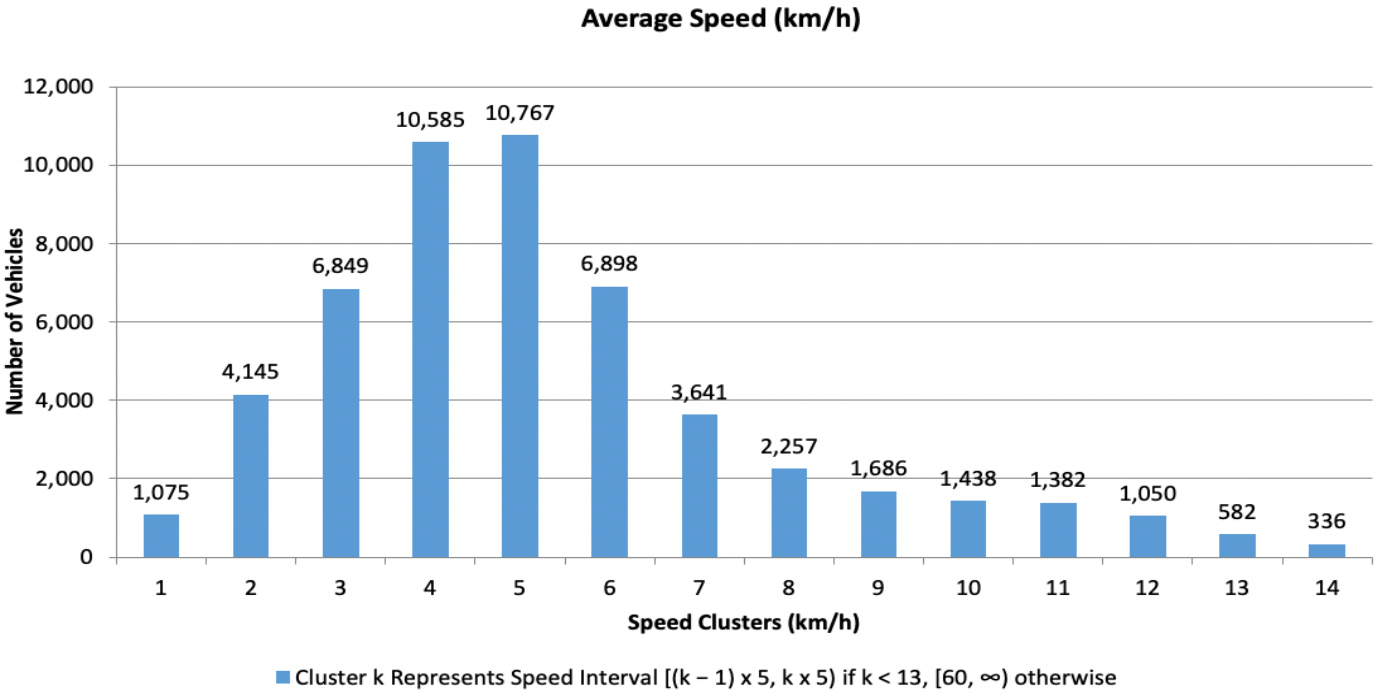


Figure 5. Speed distribution in clusters for 50 K+ simulated vehicles in Barcelona with $\delta = 1$.

4.3.2. Unbalanced Grid Topology—New York

As shown in Tables 4 and 5, the moderate congestion scenario in New York benefits from more than a 39% improvement for the total time spent in traffic, and the average vehicle speed is almost doubled (≥88%). The chart in Figure 6 shows that the traffic has a good flow because approximately 80% of the vehicles' speeds are above 10 km/h.

Table 4. Results for 50 K+ simulated vehicles in New York.

Metric	SUMO	$CASUMO_4$	$CASUMO_5$	$CASUMO_6$
s_{av}	8.10	15.26	15.65	15.68
d_{total}	377,724.01	427,372.24	434,299.56	442,962.90
t_{total}	46,604.69	28,004.75	27,753.99	28,249.23
$t_{simulation}$	7.68	7.08	6.93	7.05
$fuel_{total}$	170,255.79	112,553.47	112,128.85	113,662.42
$CO_{2_{total}}$	395,695.54	261,666.28	260,681.83	264,247.54
$countV_{peak}$	15,830	11,177	11,189	11,237

Table 5. Percentage improvement of the proposed solution for 50 K+ simulated vehicles in New York.

Metric	CASUMO$_4$	CASUMO$_5$	CASUMO$_6$
s_{av}	88.29	93.07	93.47
d_{total}	−13.14	−14.98	−17.27
t_{total}	39.91	40.45	39.39
$t_{simulation}$	7.78	9.76	8.13
$fuel_{total}$	33.89	34.14	33.24
$CO_{2_{total}}$	33.87	34.12	33.22
$countV_{peak}$	29.39	29.32	29.01

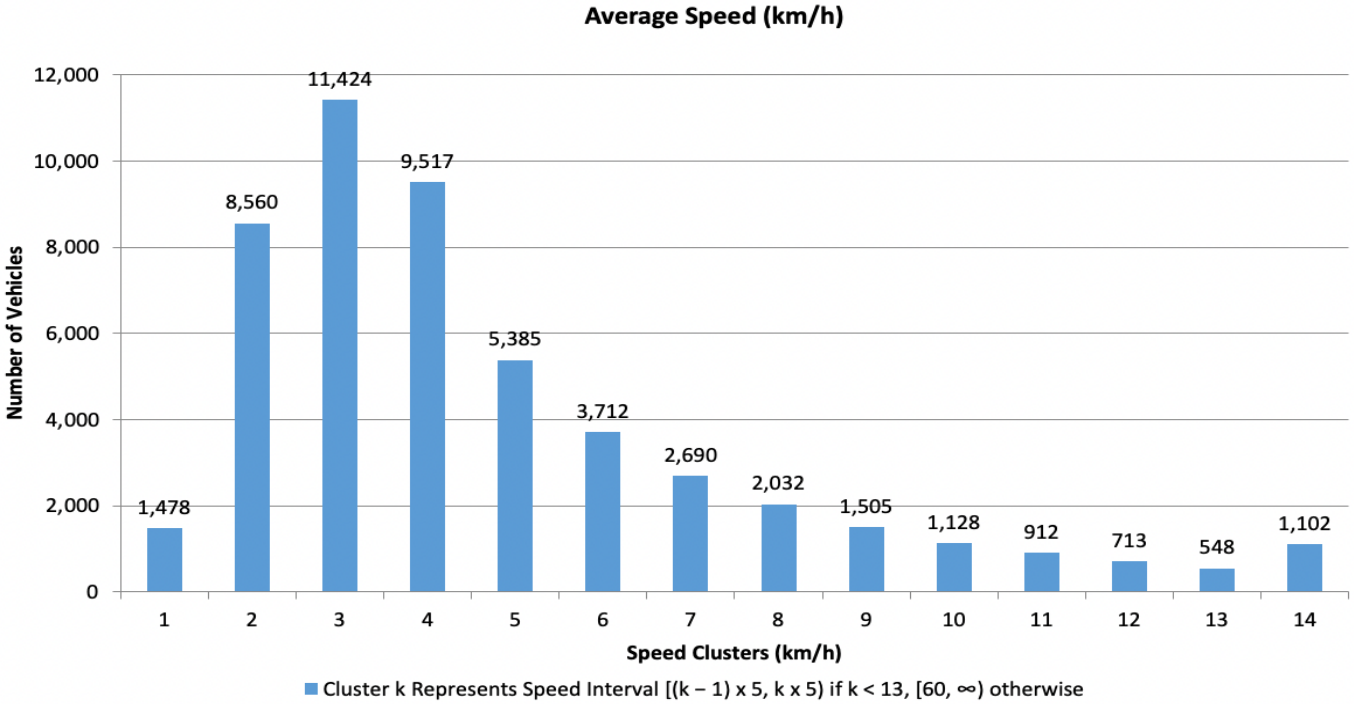


Figure 6. Speed distribution in clusters for 50 K+ simulated vehicles in New York with $\delta = 5$.

4.3.3. Historical/Irregular Topology—Bucharest

In the case of moderate congestion on a historical road network topology, the total time spent in traffic improvement by our solution was more than 42%, while the average vehicle speed was more than double that of SUMO (Tables 6 and 7). As is shown in Figure 7, more than 85% of the vehicles' speeds were at least 10 km/h.

Table 6. Results for 50 K+ simulated vehicles in Bucharest.

Metric	SUMO	CASUMO$_4$	CASUMO$_5$	CASUMO$_6$
s_{av}	8.02	17.33	17.72	18.06
d_{total}	247,890.68	306,138.41	318,841.39	333,290.65
t_{total}	30,921.99	17,666.27	17,992.65	18,452.92
$t_{simulation}$	5.23	4.31	4.34	4.16
$fuel_{total}$	104,683.61	63,475.44	65,198.86	67,108.47
$CO_{2_{total}}$	243,287.52	147,586.32	151,595.38	156,038.63
$countV_{peak}$	12,556	7564	7534	7828

Table 7. Percentage improvement of the proposed solution 50 K+ simulated vehicles in Bucharest.

Metric	CASUMO$_4$	CASUMO$_5$	CASUMO$_6$
s_{av}	116.16	121.05	125.30
d_{total}	−23.50	−28.62	−34.45
t_{total}	42.87	41.81	40.32
$t_{simulation}$	17.55	16.98	20.48
$fuel_{total}$	39.36	37.72	35.89
$CO_{2_{total}}$	39.34	37.69	35.86
$countV_{peak}$	39.76	40.00	37.66

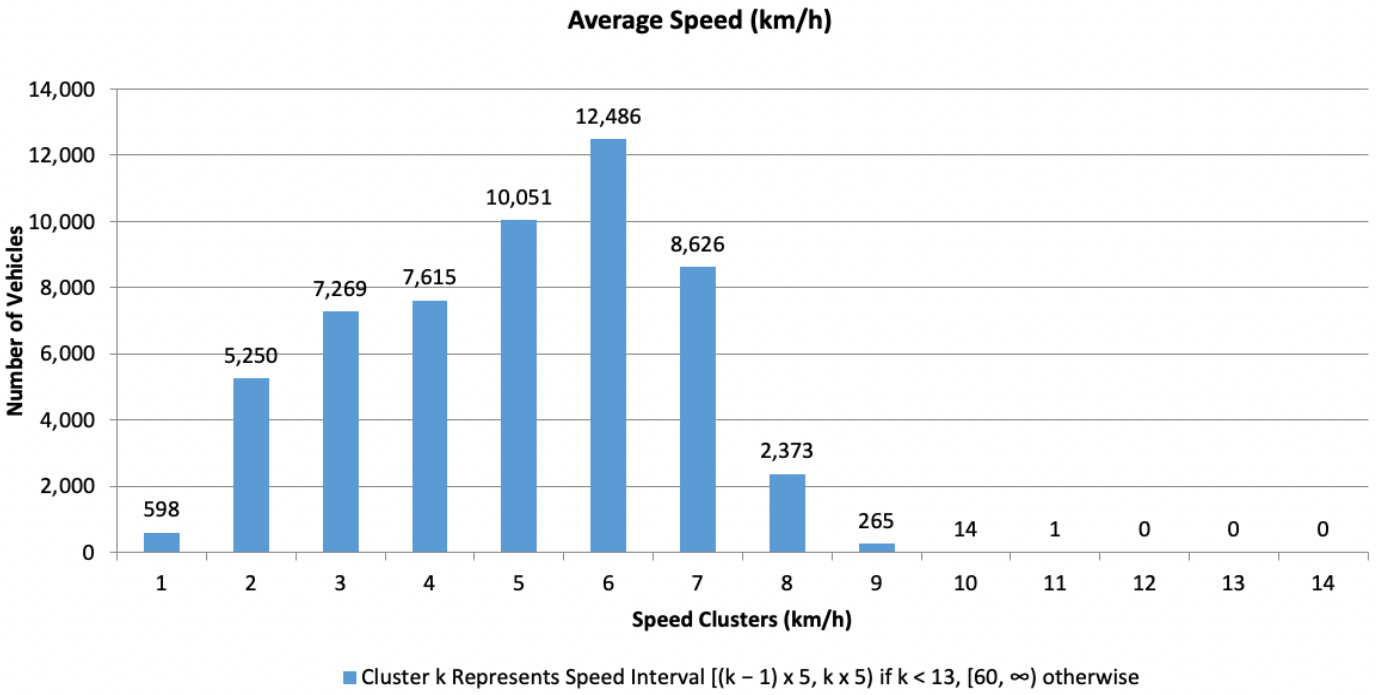


Figure 7. Speed distribution in clusters for 50 K+ simulated vehicles in Bucharest with $\delta = 4$.

4.3.4. Hybrid Topology—Tokyo

The data in Tables 8 and 9 shows that a hybrid and rich topology, such as the road network in Tokyo, is preferred by our solution, which reduces the total time spent in traffic by more than 69%, whereas the average vehicle speed is more than four times faster than the average vehicles speed in SUMO. In this case, as shown in Figure 8, more than 90% of the vehicles had speeds above 10 km/h.

Table 8. Results for 50 K+ simulated vehicles in Tokyo.

Metric	SUMO	CASUMO$_7$	CASUMO$_8$	CASUMO$_9$
s_{av}	4.16	17.04	17.55	17.78
d_{total}	264,339.49	333,266.25	338,454.75	343,217.67
t_{total}	63,536.53	19,556.52	19,287.23	19,301.42
$t_{simulation}$	9.13	5.48	5.26	5.41
$fuel_{total}$	215,887.96	83,254.05	82,873.17	83,220.06
$CO_{2_{total}}$	501,665.13	193,577.42	192,694.59	193,503.31
$countV_{peak}$	20,008	7767	7666	7681

Table 9. Percentage improvement of the proposed solution for 50 K+ simulated vehicles in Tokyo.

Metric	CASUMO$_7$	CASUMO$_8$	CASUMO$_9$
s_{av}	309.60	321.79	327.41
d_{total}	−26.08	−28.04	−29.84
t_{total}	69.22	69.64	69.62
$t_{simulation}$	40.00	42.46	40.82
$fuel_{total}$	61.44	61.61	61.45
$CO_{2_{total}}$	61.41	61.59	61.43
$countV_{peak}$	61.18	61.69	61.61

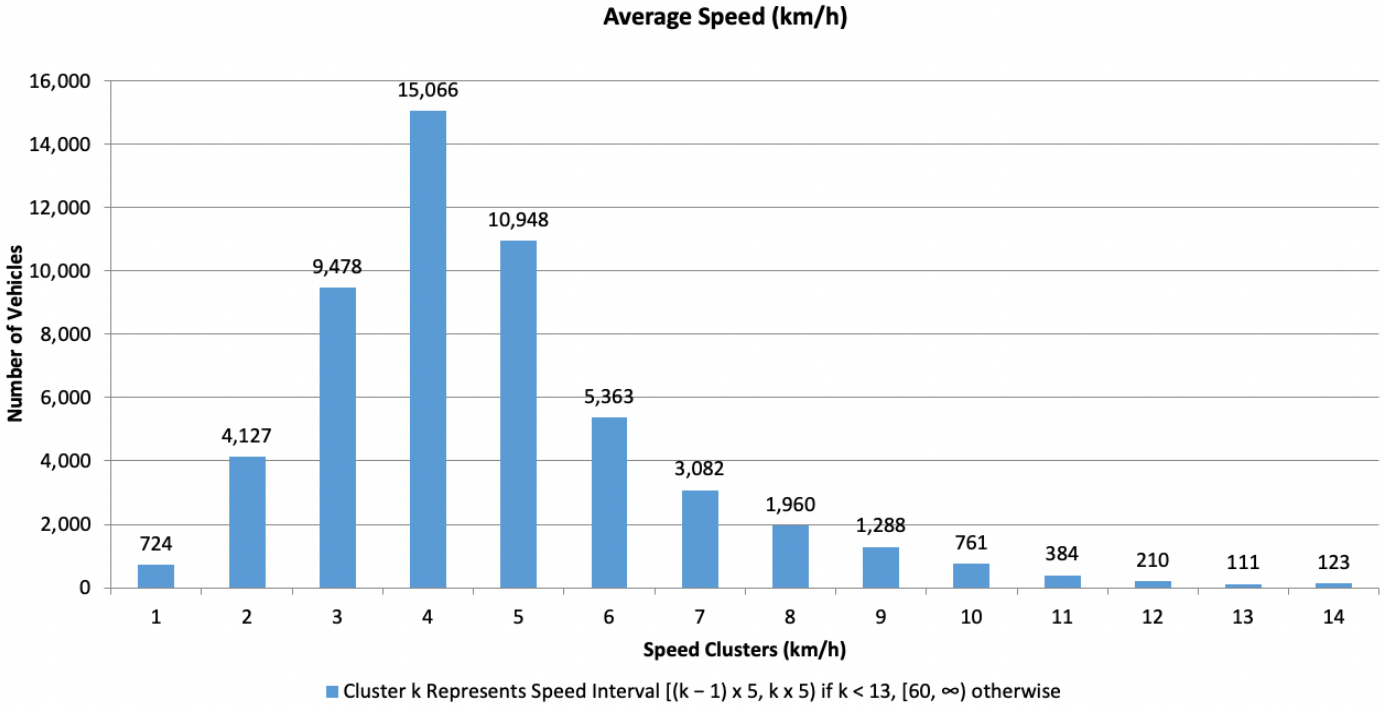


Figure 8. Speed distribution in clusters for 50 K+ simulated vehicles in Tokyo with $\delta = 8$.

4.4. Scenarios with 100 K+ Simulated Vehicles

The δ value used to obtain the best t_{total} improvement value for the 50 K+ scenarios was also used to simulate high congestion by running 100 K+ vehicles in the same four urban areas. For each urban region, a table that contains the absolute value for all the above defined metrics is presented in addition to the percentage improvement in comparison with SUMO. In addition, a chart with speed distribution in clusters is presented for each scenario.

4.4.1. Grid Topology—Barcelona

In the case of high congestion, a grid topology that follows some rules can still support an improvement of more than 32% for the total time spent in traffic, whereas the average vehicle speed improvement reaches more than 65% (see Table 10). As shown in Figure 9, because the congestion level is high, approximately 47% of the vehicles have speeds below 10 km/h, whereas only approximately 17.6% of the vehicles have the speeds below 5 km/h.

Table 10. Results and improvement of the proposed solution for 100 K+ simulated vehicles in Barcelona.

Metric	SUMO	$CASUMO_1$	% Improvement
s_{av}	5.38	8.9	65.42
d_{total}	640,470.58	714,653.69	−11.58
t_{total}	119,089.57	80,332.57	32.54
$t_{simulation}$	9.17	6.55	28.61
$fuel_{total}$	382,953.21	261,718.68	31.66
$CO_{2_{total}}$	889,893.34	608,289.76	31.64
$countV_{peak}$	38,121	30,989	18.71

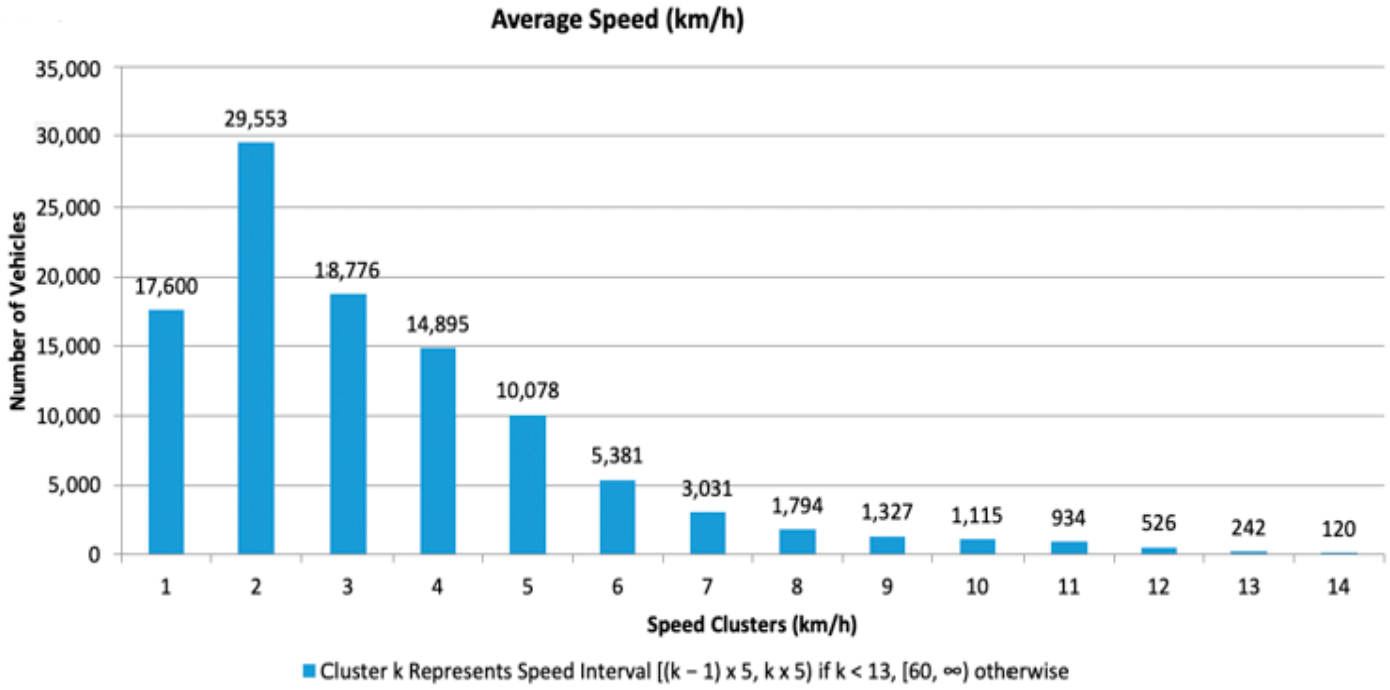


Figure 9. Speed distribution in clusters for 100 K+ simulated vehicles in Barcelona with $\delta = 1$.

4.4.2. Unbalanced Grid Topology—New York

The high congestion scenario in New York shows that, in addition to grid topology, it is worth having a constant distribution of the grids. As can be observed from Table 11, because of the unbalanced distribution, the amount of time in traffic is decreased only with 9%, whereas the speed has a 37% improvement. The unbalanced distribution of the grid topology also impacts the vehicle speeds, as shown in Figure 10: approximately 75% of the simulated vehicles have speeds below 10 km/h and 36% of the simulated vehicles have speeds below 5 km/h.

4.4.3. Historical/Irregular Topology—Bucharest

In the case of a high congestion on a historical/irregular road network topology in Bucharest, we reach almost the same values as for New York: a total time spent in traffic improvement of approximately 10% and an average vehicle speed improvement of approximately 55% (Table 12). As in the New York case, the historical/irregular road network topology impact on the vehicle speeds is high. As shown in Figure 11, approximately 65% of the simulated vehicles had speeds below 10 km/h and more than 32% of the simulated vehicles had speeds below 5 km/h.

Table 11. Results and improvement of the proposed solution for 100 K+ simulated vehicles in New York.

Metric	SUMO	$CASUMO_5$	% Improvement
s_{av}	4.39	6.03	37.31
d_{total}	798,456.07	989,623.06	−23.94
t_{total}	181,704.51	164,018.81	9.73
$t_{simulation}$	10.75	10.59	1.46
$fuel_{total}$	591,085.62	553,161.89	6.42
$CO_{2_{total}}$	1,373,525.56	1,285,547.93	6.41
$countV_{peak}$	44,273	49,181	−11.09

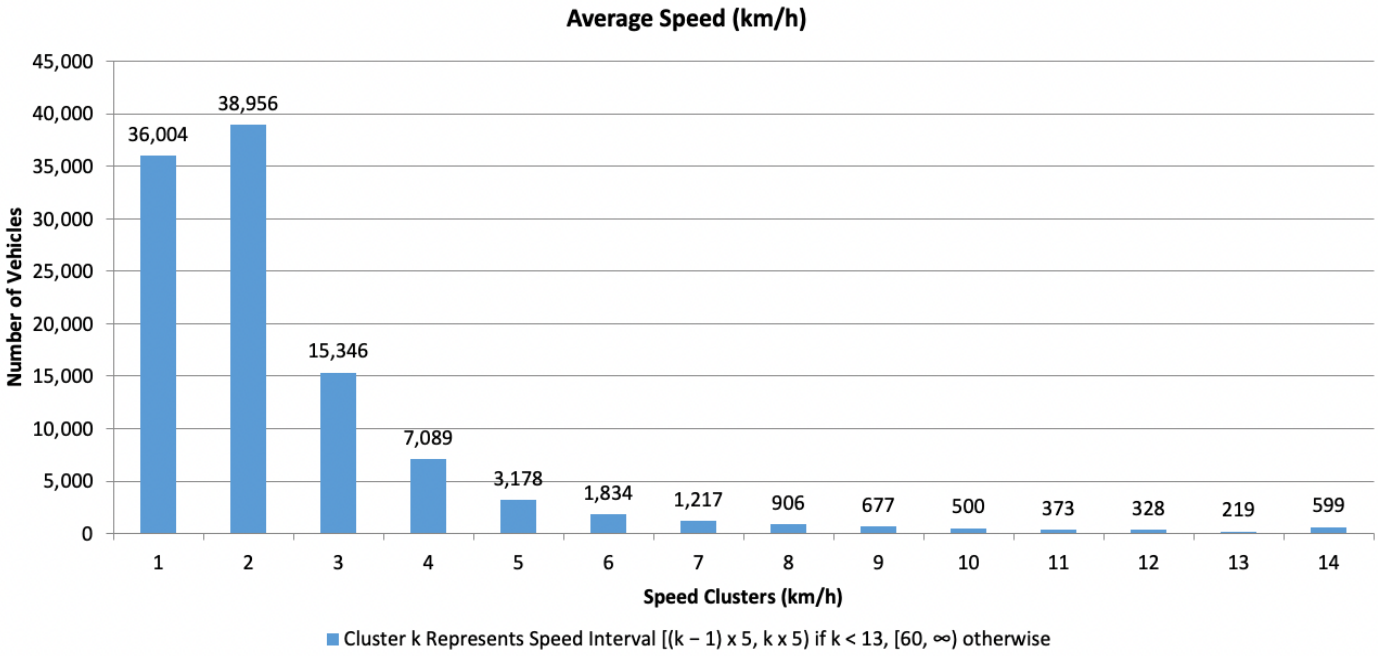


Figure 10. Speed distribution in clusters for 100 K+ simulated vehicles in New York with $\delta = 5$.

Table 12. Results and improvement of the proposed solution for 100 K+ simulated vehicles in Bucharest.

Metric	SUMO	$CASUMO_4$	% Improvement
s_{av}	4.05	6.28	55.18
d_{total}	496,049.6	691,303.51	−39.36
t_{total}	122,515.27	110,023.83	10.2
$t_{simulation}$	6.66	6.42	3.63
$fuel_{total}$	405,346.45	364,970.74	9.96
$CO_{2_{total}}$	941,818.95	848,163.21	9.94
$countV_{peak}$	39,815	41,607	−4.5

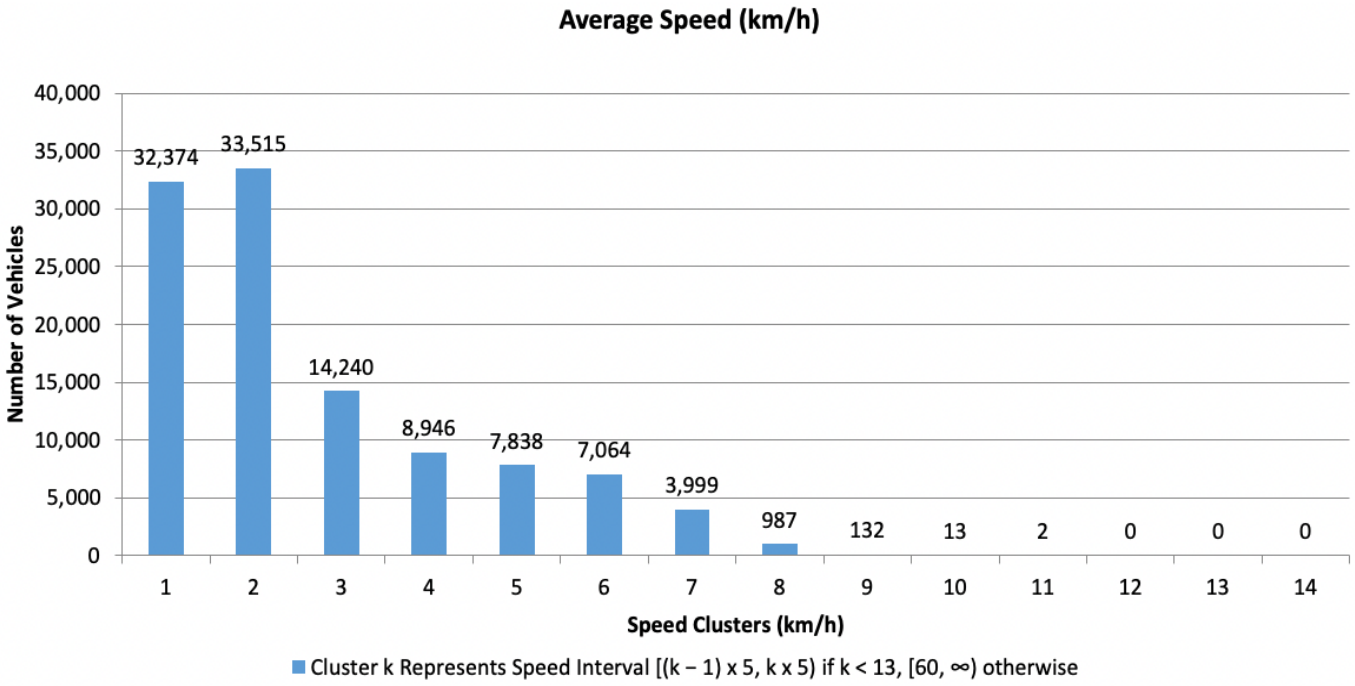


Figure 11. Speed distribution in clusters for 100 K+ simulated vehicles in Bucharest with $\delta = 4$.

4.4.4. Hybrid Topology—Tokyo

Owing to the very rich road network of the hybrid topology of Tokyo, in the case of high congestion, we obtained almost the same results as for the grid topology in Barcelona, which is approximately 32% improvement for total time spent in traffic and approximately a 104% increase in speed (see Table 13). The speed increase was higher than that in Barcelona because the routes are longer due to the multiple highway alternatives used during the route computation. The chart in Figure 12 shows that more than 55% of the simulated vehicles have speeds below 10 km/h, whereas approximately 17% of the simulated vehicles have speeds below 5 km/h. This shows that our solution provides a better overall experience on Barcelona's topology (in comparison with Tokyo) in the case of high congestion.

4.5. Barcelona Free Flow

Barcelona has the most idealistic road network topology, and we obtained very good results for 50 K+ and 100 K+ simulated vehicles. Therefore, we ran our free flow test on a selected region in Barcelona. Running SUMO with very few simulated vehicles (around 100), we found that the maximum average vehicle speed in free flow mode is almost 31.45 km/h. Considering the reports from [49], we define that, in free flow, the average vehicle's speed is at least 25 km/h. Based on this, in the selected region from Barcelona, the default SUMO can simulate at most 10 K+ vehicles and still remain in free flow mode, whereas the proposed solution can simulate at most 30 K vehicles and still remain in free flow mode. This is an improvement of 200% in the number of vehicles that can run in free flow.

4.6. Conclusions on Experimental Results

As listed in Table 1, vehicular traffic congestion was simulated by running a large number of vehicles on specific road networks multiple times. To the best of our knowledge, our solution simulates vehicular traffic congestion by covering the most comprehensive set of road network topologies (four different urban regions). We generated moderate and high traffic congestion, reaching 49 K (the most) concurrent vehicles running simultaneously on a road network. In addition, we generated free-flow scenarios for the evaluation.

Table 13. Results and improvement of the proposed solution for 100 K+ simulated vehicles in Tokyo.

Metric	SUMO	$CASUMO_8$	% Improvement
s_{av}	3.62	7.40	104.59
d_{total}	529,354.16	727,528.96	−37.44
t_{total}	146,400.54	98,348.14	32.82
$t_{simulation}$	10.35	8.71	15.89
$fuel_{total}$	490,976.24	340,683.68	30.61
$CO_{2_{total}}$	1,140,770.46	791,859.42	30.59
$countV_{peak}$	40,895	35,742	12.60

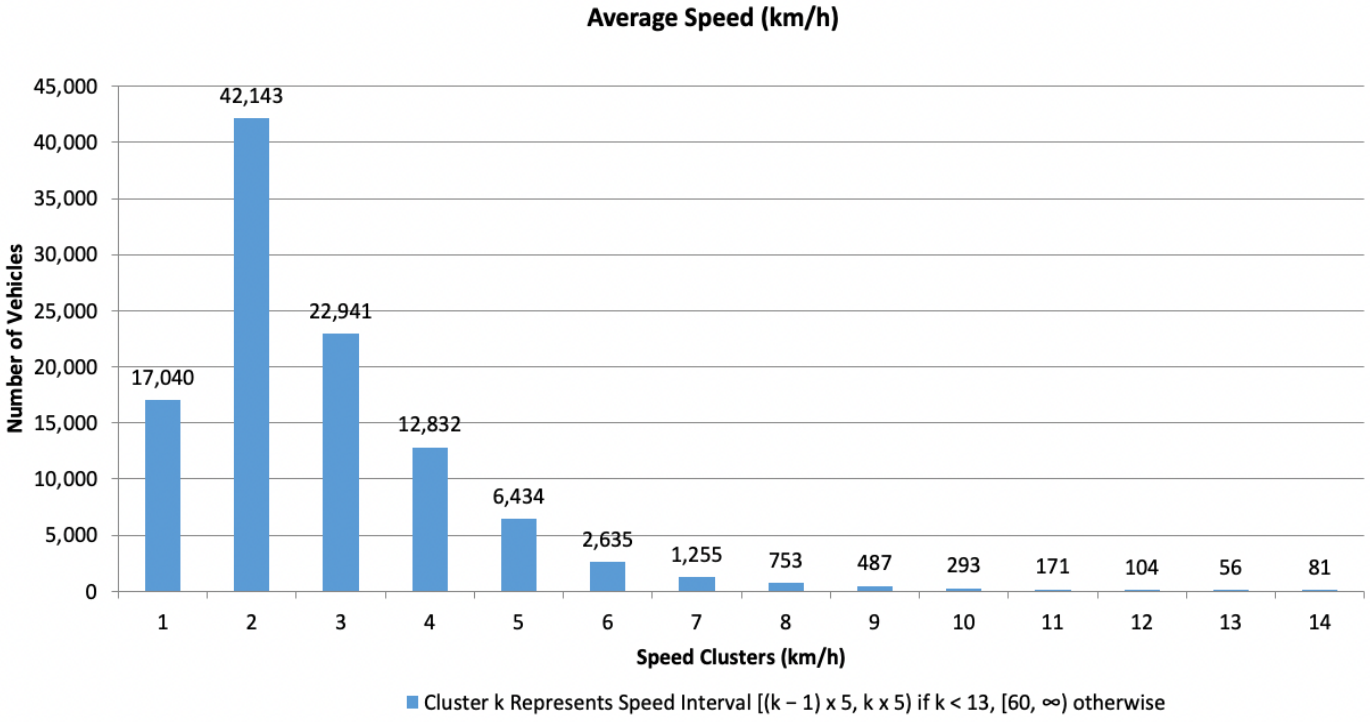


Figure 12. Speed distribution in clusters for 100 K+ simulated vehicles in Tokyo with $\delta = 8$.

Table 14 shows the maximum percentage improvement provided by the aforementioned vehicular traffic congestion reduction or avoidance solutions in urban areas. The percentage improvement was compared with the original approach, in which each congestion reduction solution was applied.

Based on the above, to the best of our knowledge, our solution tested the most comprehensive set of vehicular traffic congestion scenarios and compared it with the existing solutions, obtaining the best percentage improvement for time spent in traffic, fuel consumption, and CO_2 emissions reduction.

Table 14. Comparison of vehicular traffic congestion reduction solutions.

Solution	t_{total}%	$Fuel_{total}$%	CO_2%
CAVTTM	**69**	**61**	**61**
FOXS [45]	2.6	N/A	N/A
SOPHIA [46]	15	N/A	25.95
Re-RouTE [47]	65	N/A	N/A
Stan [8]	2.6	N / A	N/A
Elouni [50]	41	20	N/A
Chunjiang [51]	31	N/A	N/A
CACC [52]	48.3	N/A	N/A

N/A: not applicable.

5. Conclusions and Future Research

Considering the vehicular transportation challenges in urban areas due to congestion, in this paper, we proposed a novel, effective, scalable and reliable vehicular traffic congestion avoidance solution (CAVTTM). We introduced a new vehicular traffic thresholding mechanism that is used to reduce and avoid traffic congestion on a comprehensive set of road network topologies. In the proposed solution, traffic modelling on the road network was performed using the KEP tree data structure we proposed in [8]. By integrating the proposed vehicular traffic thresholding mechanism into the route computation algorithm from SUMO, this paper reports the following achievements:

- Using the proposed congestion avoidance solution (CAVTTM), we simulated 26 vehicular traffic scenarios that ran on a comprehensive set of road network topologies in four urban areas: Barcelona, Bucharest, New York and Tokyo;
- Generated moderate and high traffic congestion by running between 50 K+ and 100 K vehicles for each simulation and reaching up to 49 K concurrent vehicles that run on a road network topology at a specific moment;
- Through the vehicular traffic thresholding mechanism, for the scenarios with 100 K+ vehicles, we reduced the time spent in traffic by more than 32% and CO_2 emissions by more than 31%, whereas, for the scenarios with 50 K+ vehicles, we reduced the time spent in traffic by more than 69% and CO_2 emissions with more than 61%;
- Using a vehicular traffic thresholding mechanism, we balanced traffic in Barcelona to improve the number of simulated vehicles in free flow by 200%;
- The proposed CAVTTM is an effective, scalable and reliable solution to traffic congestion that can be used in most road network topologies.

A noteworthy work for the future is to introduce into CAVTTM routes that are not generated using the vehicular traffic thresholding mechanism but are generated in an individualistic manner (e.g., fastest routes at a specific moment) and to observe the evolution of the entire vehicular context. From a more practical perspective, for further work, we can focus on officially integrating the proposed vehicular traffic thresholding mechanism in the SUMO's route computation algorithm so that the community can benefit from simulated routes that balance traffic and avoid congestion. Furthermore, it would be very valuable to integrate the proposed vehicular traffic thresholding mechanism in the field, by adapting the route computation algorithm of an existing navigation solution to use the thresholding levels and KEP tree to model traffic data. When used by users in a specific urban area, such a solution can provide valuable traffic data to be evaluated. This study is limited to predicting and avoiding vehicular traffic congestion in the context of large-scale simulated vehicular traffic on a comprehensive set of road network topologies.

Author Contributions: Conceptualization, I.S., D.A.G., P.I.D. and R.P.; methodology, I.S., D.A.G., P.I.D. and R.P.; software, I.S., D.A.G., P.I.D. and R.P.; validation, I.S., D.A.G., P.I.D. and R.P.; formal analysis, I.S., D.A.G., P.I.D. and R.P.; investigation, I.S., D.A.G., P.I.D. and R.P.; resources, I.S., D.A.G., P.I.D. and R.P.; data curation, I.S., D.A.G., P.I.D. and R.P.; writing—original draft preparation, I.S., D.A.G., P.I.D. and R.P.; writing—review and editing, I.S., D.A.G., P.I.D. and R.P.; visualization, I.S., D.A.G., P.I.D. and R.P.; supervision, I.S. and R.P.; project administration, I.S. and R.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Urban Mobility Report. Available online: https://mobility.tamu.edu/umr/report/ (accessed on 16 April 2021).
2. Mobility and Transport. Available online: https://ec.europa.eu/transport/themes/urban/urban_mobility_en (accessed on 16 April 2021).
3. Chu, K.; Saigal, R.; Saitou, K. Real-Time Traffic Prediction and Probing Strategy for Lagrangian Traffic Data. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 497–506. [CrossRef]
4. Mitrovic, N.; Asif, M.T.; Dauwels, J.; Jaillet, P. Low-Dimensional Models for Compressed Sensing and Prediction of Large-Scale Traffic Data. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2949–2954. [CrossRef]
5. Liu, J.; Wan, J.; Jia, D.; Zeng, B.; Li, D.; Hsu, C.; Chen, H. High-Efficiency Urban Traffic Management in Context-Aware Computing and 5G Communication. *IEEE Commun. Mag.* **2017**, *55*, 34–40. [CrossRef]
6. Guleng, S.; Wu, C.; Liu, Z.; Chen, X. Edge-Based V2X Communications With Big Data Intelligence. *IEEE Access* **2020**, *8*, 8603–8613. [CrossRef]
7. Mitrovic, N.; Narayanan, A.; Asif, M.T.; Rauf, A.; Dauwels, J.; Jaillet, P. On Centralized and Decentralized Architectures for Traffic Applications. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1988–1997. [CrossRef]
8. Stan, I.; Suciu, V.; Potolea, R. Scalable Data Model for Traffic Congestion Avoidance in a Vehicle to Cloud Infrastructure. *Sensors* **2021**, *21*, 5074. [CrossRef]
9. Stan, I.; Ungur, B.P.; Potolea, R. Augmented Van Emde Boas Tree for Connected Vehicles Traffic Modeling. In Proceedings of the 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 28–30 October 2021; pp. 199–208. [CrossRef]
10. Chen, C.L.P.; Zhou, J.; Zhao, W. A Real-Time Vehicle Navigation Algorithm in Sensor Network Environments. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1657–1666. [CrossRef]
11. Liao, T.Y.; Hu, T.Y. An object-oriented evaluation framework for dynamic vehicle routing problems under real-time information. *Expert Syst. Appl.* **2011**, *38*, 12548 – 12558. [CrossRef]
12. Gendreau, M.; Potvin, J.Y.; Bräumlaysy, O.; Hasle, G.; Løkketangen, A., *Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography;* Springer: New York, NY, USA, 2008; Volume 43, pp. 143–169. [CrossRef]
13. Stan., I.; Suciu., V.; Potolea., R. Routing Algorithms in Connected Cars Context. In Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Vienna, Austria, 17–19 September 2019; Volume 2, pp. 290–297. [CrossRef]
14. Bakar, N.A.A.; Adi, A.F.N.; Majid, M.A.; Adam, K.; Younis, Y.M.; Fakhreldin, M. The Simulation on Vehicular Traffic Congestion Using Discrete Event Simulation (DES): A Case Study. In Proceedings of the 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, 18–20 November 2018; pp. 1–6.
15. Fernandes, R.; Vieira, F.; Ferreira, M. VNS: An integrated framework for vehicular networks simulation. In Proceedings of the 2012 IEEE Vehicular Networking Conference (VNC), Seoul, Republic of Korea, 14–16 November 2012; pp. 195–202. [CrossRef]
16. Elbery, A.; Rakha, H.; ElNainay, M. Vehicular Communication and Mobility Sustainability: the Mutual Impacts in Large-scale Smart Cities. *arXiv* **2019**, arXiv:1908.08229.
17. Elbery, A.; Rakha, H.A.; ElNainay, M. Large-Scale Modeling of VANET and Transportation Systems. In *Proceedings of the Traffic and Granular Flow '17*; Hamdar, S.H., Ed.; Springer International Publishing: Cham, Switzerland, 2019; pp. 517–526.
18. Farag, M.M.G.; Rakha, H.A.; Mazied, E.A.; Rao, J. INTEGRATION Large-Scale Modeling Framework of Direct Cellular Vehicle-to-All (C-V2X) Applications. *Sensors* **2021**, *21*, 2127. [CrossRef]
19. Stan, I.; Toderici, D.; Potolea, R. Segment Trees based Traffic Congestion Avoidance in Connected Cars Context. In Proceedings of the 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 6–8 September 2018; pp. 137–143.

20. Stan, I.; Ghisa, R.; Potolea, R. Urban Traffic Simulation Methodology for Connected Vehicles Congestion Avoidance. In Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services (iiWAS '20), Chiang Mai, Thailand, 30 November–2 December 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 305–312. [CrossRef]
21. Stan, I.; Suciu, V.; Potolea, R. Smart Driving Methodology for Connected Cars. In Proceedings of the 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 9–11 October 2019; pp. 608–613. [CrossRef]
22. Comert, G.; Bezuglov, A. An Online Change-Point-Based Model for Traffic Parameter Prediction. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1360–1369. [CrossRef]
23. Keyvan-Ekbatani, M.; Kouvelas, A.; Papamichail, I.; Papageorgiou, M. Exploiting the fundamental diagram of urban networks for feedback-based gating. *Transp. Res. Part B Methodol.* **2012**, *46*, 1393–1403. [CrossRef]
24. Geroliminis, N.; Daganzo, C.F. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transp. Res. Part B Methodol.* **2008**, *42*, 759 – 770. [CrossRef]
25. Salamanis, A.; Kehagias, D.D.; Filelis-Papadopoulos, C.K.; Tzovaras, D.; Gravvanis, G.A. Managing Spatial Graph Dependencies in Large Volumes of Traffic Data for Travel-Time Prediction. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1678–1687. [CrossRef]
26. Bandaragoda, T.; De Silva, D.; Kleyko, D.; Osipov, E.; Wiklund, U.; Alahakoon, D. Trajectory clustering of road traffic in urban environments using incremental machine learning in combination with hyperdimensional computing. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 1664–1670.
27. Salazar-Carrillo, J.; Torres-Ruiz, M.; Davis, C.A.; Quintero, R.; Moreno-Ibarra, M.; Guzmán, G. Traffic Congestion Analysis Based on a Web-GIS and Data Mining of Traffic Events from Twitter. *Sensors* **2021**, *21*, 2964. [CrossRef]
28. Pan, J.; Khan, M.A.; Popa, I.S.; Zeitouni, K.; Borcea, C. Proactive Vehicle Re-routing Strategies for Congestion Avoidance. In Proceedings of the 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems, Hangzhou, Zhejiang, 16–18 May 2012; pp. 265–272.
29. Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Krajzewicz, D. SUMO – Simulation of Urban MObility: An Overview. In Proceedings of the Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain, 23–29 October 2011; Volume 2011.
30. Wegener, A.; Piorkowski, M.; Raya, M.; Hellbrück, H.; Fischer, S.; Hubaux, J.P. TraCI: An Interface for Coupling Road Traffic and Network Simulators. In Proceedings of the 11th Communications and Networking Simulation Symposium (CNS'08), New York, NY, USA, 14 April 2008. [CrossRef]
31. Jiang, Z.; Wu, J.; Sabatino, P. GUI: GPS-Less Traffic Congestion Avoidance in Urban Areas with Inter-Vehicular Communication. In Proceedings of the 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, Philadelphia, PA, USA , 28–30 October 2014; pp. 19–27.
32. Cárdenas-Benítez, N.; Aquino-Santos, R.; Magaña-Espinoza, P.; Aguilar-Velazco, J.; Edwards-Block, A.; Medina Cass, A. Traffic Congestion Detection System through Connected Vehicles and Big Data. *Sensors* **2016**, *16*, 599. [CrossRef]
33. Wang, S.; Djahel, S.; McManis, J. A Multi-Agent based vehicles re-routing system for unexpected traffic congestion avoidance. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 2541–2548.
34. Lindorfer, M.; Mecklenbräuker, C.; Ostermayer, G. Modeling the Imperfect Driver: Incorporating Human Factors in a Microscopic Traffic Model. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1–15. [CrossRef]
35. Wu, Y.; Tan, H.; Qin, L.; Ran, B.; Jiang, Z. A hybrid deep learning based traffic flow prediction method and its understanding. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 166–180. [CrossRef]
36. Olayode, I.O.; Tartibu, L.K.; Okwu, M.O. Prediction and modeling of traffic flow of human-driven vehicles at a signalized road intersection using artificial neural network model: A South African road transportation system scenario. *Transp. Eng.* **2021**, *6*, 100095. [CrossRef]
37. Qi, L.; Zhou, M.; Luan, W. Impact of Driving Behavior on Traffic Delay at a Congested Signalized Intersection. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1882–1893. [CrossRef]
38. Qi, L.; Zhou, M.; Luan, W. A Dynamic Road Incident Information Delivery Strategy to Reduce Urban Traffic Congestion. *Acta Autom. Sin.* **2018**, *5*, 934–945. [CrossRef]
39. Qi, L.; Zhou, M.; Luan, W. A Two-level Traffic Light Control Strategy for Preventing Incident-Based Urban Traffic Congestion. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 13–24. [CrossRef]
40. Qi, L.; Zhou, M.; Luan, W. Emergency Traffic-Light Control System Design for Intersections Subject to Accidents. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 170–183. [CrossRef]
41. Backfrieder, C.; Lindorfer, M.; Mecklenbraeuker, C.F.; Ostermayer, G. Impact of Varying Penetration Rate of Intelligent Routing Capabilities on Vehicular Traffic Flow. In Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, Canada, 24–27 September 2017; pp. 1–6.
42. Backfrieder, C.; Ostermayer, G. Modeling a Continuous and Accident-Free Intersection Control for Vehicular Traffic in TraffSim. In Proceedings of the 2014 European Modelling Symposium, Pisa, Italy, 21–23 October 2014; pp. 332–337.
43. Backfrieder, C.; Ostermayer, G.; Mecklenbräuker, C.F. Increased Traffic Flow Through Node-Based Bottleneck Prediction and V2X Communication. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 349–363. [CrossRef]

44. Blazek, T.; Mecklenbräuker, C.F.; Backfrieder, C.; Ostermayer, G. Improving communication reliability in intelligent transport systems through cooperative driving. In Proceedings of the 2017 10th IFIP Wireless and Mobile Networking Conference (WMNC), Valencia, Spain, 25–27 September 2017; pp. 1–6.
45. Brennand, C.A.R.L.; Filho, G.P.R.; Maia, G.; Cunha, F.; Guidoni, D.L.; Villas, L.A. Towards a Fog-Enabled Intelligent Transportation System to Reduce Traffic Jam. *Sensors* **2019**, *19*, 3916. [CrossRef]
46. Akabane, A.T.; Immich, R.; Pazzi, R.W.; Madeira, E.R.M.; Villas, L.A. Exploiting Vehicular Social Networks and Dynamic Clustering to Enhance Urban Mobility Management. *Sensors* **2019**, *19*, 3558. [CrossRef]
47. Guidoni, D.L.; Maia, G.; Souza, F.S.H.; Villas, L.A.; Loureiro, A.A.F. Vehicular Traffic Management Based on Traffic Engineering for Vehicular Ad Hoc Networks. *IEEE Access* **2020**, *8*, 45167–45183. [CrossRef]
48. Lee, S.; Younis, M.; Murali, A.; Lee, M. Dynamic Local Vehicular Flow Optimization Using Real-Time Traffic Conditions at Multiple Road Intersections. *IEEE Access* **2019**, *7*, 28137–28157. [CrossRef]
49. TomTom, 2022. Available online: https://www.tomtom.com/en_gb/traffic-index/barcelona-traffic/ (accessed on 23 May 2022).
50. Elouni, M.; Abdelghaffar, H.M.; Rakha, H.A. Adaptive Traffic Signal Control: Game-Theoretic Decentralized vs. Centralized Perimeter Control. *Sensors* **2021**, *21*, 274. [CrossRef]
51. Wu, C.; Zhou, S.; Xiao, L. Dynamic Path Planning Based on Improved Ant Colony Algorithm in Traffic Congestion. *IEEE Access* **2020**, *8*, 180773–180783. [CrossRef]
52. Cao, Z.; Lu, L.; Chen, C.; Chen, X. Modeling and Simulating Urban Traffic Flow Mixed With Regular and Connected Vehicles. *IEEE Access* **2021**, *9*, 10392–10399. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.