CANS: Context-Aware Traffic Estimation and Navigation System

Azam Ramazani, Hamed Vahdat-Nejad

PerLab, Faculty of Electrical and Computer Engineering, University of Birjand, Iran

ramazani_azam@birjand.ac.ir, vahdatnejad@birjand.ac.ir

Abstract-Acquiring real-time traffic information is a basic requirement for dynamic vehicular navigation systems. The majority of the current navigation systems are based on static traffic information. Building on Mobile Crowd Sensing technology, we propose CANS (**C**ontext-**A**ware Traffic Estimation and **N**avigation **S**ystem), a context-aware system that can estimate traffic state without any requirement for expensive infrastructure. Using only available equipment, it can provide dynamic navigation service to drivers. The proposed system consists of three main components including local traffic estimation, global traffic aggregation, and navigation. In this system, vehicles estimate local traffic state using vehicular contextual information including speed and acceleration by relying on fuzzy logic, and transmit the information to the urban server. The server integrates the received local traffic information from different vehicles and estimates the global traffic state, providing the traffic-aware navigation system to drivers. CANS performance is evaluated for an urban scenario in a traffic flow in Birjand, Iran. The experiment is conducted for evaluating CANS in both traffic congestion estimation as well as navigation. The results show an accurate estimation of traffic states along urban roads. Compared with previous approaches, CANS overrides them for its reduced travel time.

Keywords: Vehicular Ad-hoc Network, Traffic Congestion Estimation, Navigation, Pervasive Computing, Mobile Crowd Sensing

1- Introduction

Vehicular Ad-hoc Network is a branch of Mobile Ad-hoc Networks (MANETs) and a part of the intelligent transportation systems, which has emerged thanks to the advances in wireless technology and automobile industry [1-5]. VANET allows vehicles to communicate directly with each other, known as vehicle-to-vehicle communication (V2V). Vehicles can also communicate with road-side units via vehicle-to-infrastructure communication (V2I) [6-8]. VANETs have provided the required infrastructure for a wide range of transportation applications [9].

Nowadays, with the increase in urban traffic, a main concern of drivers is to find the route with the shortest travel time. The development of an efficient navigation system for urban roads that can locate an optimum route from the starting point to the destination is an important issue in intelligent transportation systems. This relies on important factors such as the length of the route, current traffic state, and conditions of the route including maximum speed limit and the number of traffic lights along the route. The more intelligently these factors are selected, the better the selection of the optimized route. Some of these factors are static e.g., length of the roads and speed limit. However, some are dynamic and change over time, e.g. traffic level. Therefore, two key issues need to be taken into consideration when developing a navigation system based on dynamic traffic information: the way to acquire dynamic traffic information, and the selection of the shortest route in terms of travel time.

Today, Traffic level of streets is usually obtained via deploying special purpose infrastructure including video cameras [10-12] and loop detectors [13-15]. The drawbacks of this mechanism include low accuracy (because of spotted data collection), incomprehensive spatial coverage, as well as high costs for infrastructure deployment and maintenance [16]. Alternatively, probe vehicles have been utilized to periodically sense and report their position, speed, and travel time. Hence, a specific server gathers these data and estimates the traffic congestion level, accordingly [17-19]. In fact, this approach is based on the dynamic collection of traffic data by certain mobile vehicles distributed across the road network. While these techniques have the advantage of widespread spatial and temporal coverage and high accuracy, they require probe vehicles for traffic information collection, which is indeed costly. At the same time, some studies have used the vehicular networks to detect traffic conditions [20-23]. On the other hand, most current approaches to navigation, ignore traffic information, or solely navigate on static traffic information. In the latter approach, the amount of traffic is considered invariant at all times. Therefore, it cannot reflect the emergency events such as accidents or temporary events like road construction, which lead to different traffic congestion.

In light of the significant advances in information technology, the new idea of context-aware systems has emerged as an efficient approach to improve the area of transportation. Context-aware systems continuously monitor the environment and adapt their actions to the environmental situation. Context is defined as "any information that can be used to characterize the situation of an entity" [24, 25]. This paper makes use of this key technology together with vehicular networks for dynamically estimating traffic state.

In this research, a **C**ontext-**A**ware Traffic Estimation and **N**avigation **S**ystem (CANS) is proposed. The traffic estimation part is an extension of our previous paper [26] in which a vehicle could only estimate its local traffic. This idea is now extended using Mobile Crowd Sensing technology [27, 28] such that the new system is able to estimate the global traffic of a whole region, dynamically. Moreover, CANS uses this dynamic traffic information for navigation. It provides a service to find the shortest route in terms of travel time. The proposed architecture of CANS consists of three main components including local traffic estimation, global traffic estimation, and navigation. The local traffic estimation is adopted from our previous work [26] in which vehicles act as mobile traffic sensors that measure their local traffic state. For this, each vehicle periodically computes its context information of average speed and mean absolute acceleration. Afterward, it uses them to estimate local traffic congestion by utilizing a fuzzy system. After that, each vehicle transmits the information about the measured local traffic congestion of its relevant road segment to the urban server through vehicle-to-infrastructure communication (V2I). The global traffic estimation component on the urban server aggregates the received local traffic information from the vehicles and publicizes the global traffic state of the city. The navigation component on the urban server uses the global traffic information to find the optimum route between an origin and destination pair. It estimates the cost of each street segment in terms of travel time plus the delay time at intersections and waiting time behind traffic lights. When a vehicle requests for a route to the destination, the navigation component uses the global traffic information to estimate the cost of each street segment and locate the best path using Dijkstra algorithm. Thus, the optimal route and information about, for example, travel time and length of the route, are displayed to the driver.

In summary, the contribution of the paper is twofold. First, it uses context-awareness and mobile crowdsensing to dynamically estimate local traffic and to aggregate it for being used by the navigation system. Second, it proposes the architecture of a navigation system that exploits this dynamic information to provide a path with lowest travel time. The performance of the proposed system is

evaluated in a scenario based on road network of Birjand City, Iran, using realistic traffic flow. For this purpose, one section of the city with an area of 7 km * 6 km with 362 street segments and 102 intersections is considered. The evaluation results show an accurate traffic level estimation on urban roads. In addition, the proposed navigation system is evaluated and compared with static and CATE [29] navigation systems for different origin-destination pairs. The evaluation results show that the proposed system has better performance for routing in urban network in terms of reducing the travel time.

The rest of the paper is organized as follows: Section 2 reviews related works on context-aware traffic estimation and navigation. In section 3, we describe the proposed system architecture, which involves three components for traffic state estimation and navigation. Section 4 presents the experiments accomplished to evaluate the performance of the CANS, and discusses the comparison results. Finally, conclusions and future research directions are provided in section 5.

2- Related Works

Advances in context-aware systems have motivated researchers to use this new technology as a potential approach for estimation and measurement of traffic state as well as for vehicular navigation. In this section, previous research concerning context-aware traffic estimation and navigation is reviewed.

2-1 Context-aware Traffic Estimation

Traffic congestion estimation has been studied extensively. The majority of these studies are based on deploying specific infrastructure to collect traffic data, which requires considerable cost to implement. Moreover, it suffers from incomprehensive spatial coverage and low accuracy because of spotted data collection and generalizing it. In recent years, there have been a few enquiries that have applied context-aware computing technology to estimate traffic state, which are reviewed below.

Considering the time-dependent nature of traffic flow, a road traffic prediction model has been proposed [30]. It is based on differentiating between peak and non-peak traffic periods. It uses them to train two context-aware random forests for each time period as the traffic prediction model. Nonetheless, the prediction occurs with a high error rate, and dynamic traffic information such as accidents is ignored.

Raphiphan et al. [31] have developed a distributed traffic report system to help route selection. Their study calculates traffic state for sensor-less road segments or occasions where mobile sensors' data is not available. This approach builds on combining context information and reasoning over them. To do this, the approach uses the contextual information including weather data, time of the day, day of the week, workday, school break, as well as previous traffic information to reason traffic state. It actually predicts traffic state by considering traffic patterns extracted from a set of general context. It is rather a static approach and cannot consider the impact of dynamic situations on traffic state.

In our previous paper [26], local traffic state is estimated by proposing a fuzzy system. In this system, each vehicle acts as a mobile traffic sensor and estimates its local traffic congestion using two fuzzy variables of average speed and mean absolute acceleration. This idea is extended in the current study using mobile crowd sensing technology such that the global traffic state is estimated dynamically as high-level context information by integrating the local traffic information computed by the vehicles. Later, building on this dynamic traffic information, a traffic-aware navigation system is developed.

2-2 Context-aware Navigation

Many of the available navigation systems rely on finding the shortest distance path. However, sometimes driving a longer path may take less time due to the traffic situation. Kim et al. [32] have developed Optimize Your Time (OYT) System to find the fastest subway path. The system uses the user's context information including time and current location to find the nearest station to the user. Based on the timetable of trains and the destination of the user, it displays the optimal route in terms of travel time.

Wang et al. [33] have proposed an energy-driven route planning system for fully electric vehicles. The system uses traffic data, road length, and road slope and offers a route with minimum travel time and energy consumption. In this system, each vehicle computes the traffic level on the road link as the ratio of the average speed to the road speed limit. The result is shared with other vehicles through vehicle-to-vehicle (V2V) communication. The system has been evaluated using SUMO simulator on the road network of Bradford, UK.

CATE (Computer-Assisted Traveling Environment) [29] allows vehicles to share the traffic information with their neighboring vehicles, and reroute dynamically based on the collected traffic information. In this system, each vehicle computes its local traffic state. The result is then shared with neighboring vehicles. In CATE, vehicles' travel time is considered as the degree of congestion on the road. Whenever a vehicle exits a road segment, it creates a traffic sample involving time spent by the vehicle on road link, which is disseminated to the neighboring vehicles. Using the collected traffic samples, the vehicles can then reroute dynamically. Moreover, a number of solutions are tested and evaluated for accurate estimation of the current traffic conditions on the basis of the collected traffic samples in the scenario of Portland. From among the solutions, "most recent estimate" algorithm shows a more satisfactory performance than other solutions in reducing travel time. In this algorithm, CATE selects the most recent sample as the link's weight and based on which, it navigates. The solution has a proper ability to reduce travel time as far as traffic state would not change quickly. However, if the traffic state is highly dynamic, link's weight can heavily fluctuate since the traffic samples vary. In this case, the most recent sample could cause error in estimating the current traffic state.

The CANS system does not require any specialized infrastructure or probe vehicles to collect traffic information, making it cost-effective. Mobile crowd sensing is utilized to estimate traffic state dynamically by using vehicles as traffic sources. Subsequently, the global dynamic traffic information is utilized to navigate user's destination.

3- CANS: The Proposed Framework

In this section, the architecture design of CANS is described. CANS provides a dynamic traffic estimation as well as traffic-aware navigation system for vehicles. Figure 1 shows the overall structure of CANS's architecture. The proposed system consists of three main components: local traffic estimation, global traffic estimation, and navigation. The local traffic estimation component is located on the vehicle, while the other two components are on the urban server. Below, each one of the components is explained in detail.

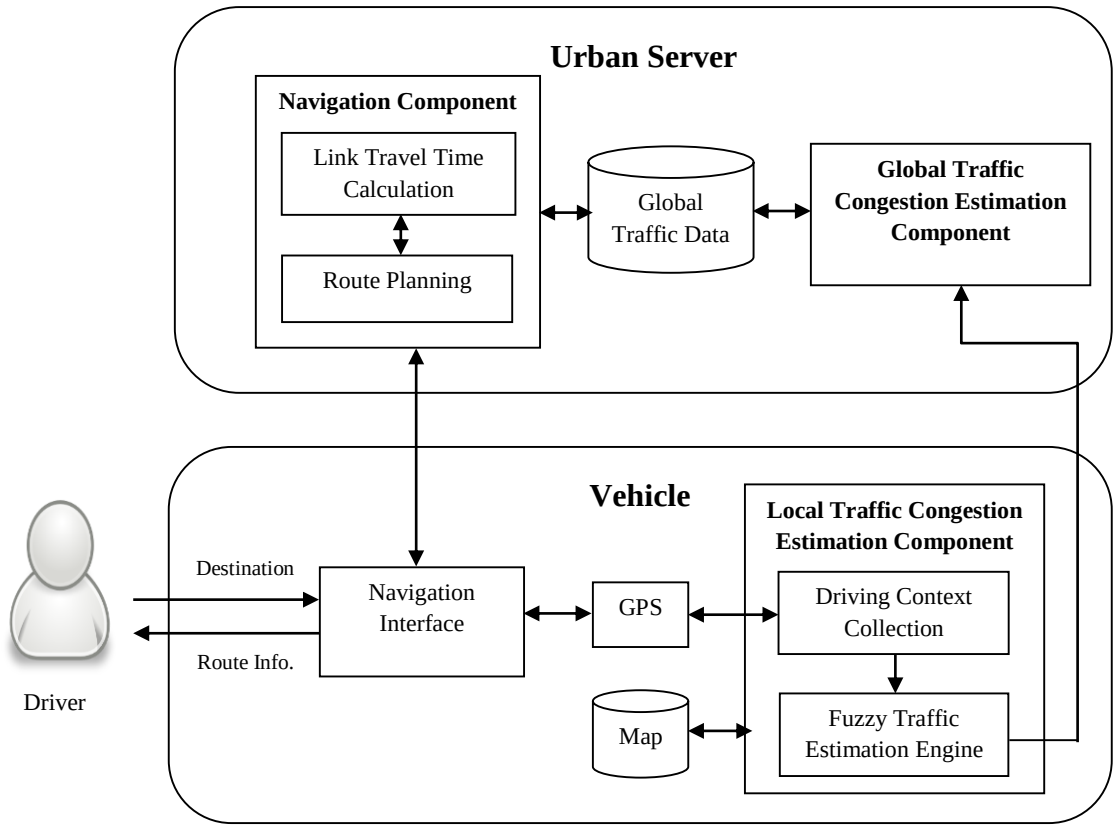

Figure 1: Architecture of CANS

3-1 Traffic Estimation

The first major functionality of CANS is traffic congestion estimation. It makes use of Crowd Sensing technology [27, 28], and obtains traffic information via a two-level scheme. At the bottom, each car inferences local traffic and acts as a sensor. At the top layer, the information generated by a crowd of cars is aggregated to the global traffic information.

3-1-1 Local Traffic Estimation

Local traffic estimation is adopted from our previous paper [26]. We briefly describe it in this subsection. As shown in Figure 1, the local traffic congestion estimation module is located on the vehicles. In fact, each vehicle is considered as a mobile sensor to obtain its local traffic information. For this, they periodically compute the average speed and mean absolute acceleration as contextual information. Afterwards, these contextual elements are regarded as two fuzzy variables. The fuzzy system aims to estimate the local traffic state as the ouptput fuzzy variable. It seems that speed of a vehicle has reverse relation with local traffic state. Therefore, the average speed is used as the first fuzzy variable and is calculated by the following formula:

$$\bar{V}_T = \frac{v_1 + v_2 + \ldots + v_m}{m} \qquad (1)$$

,Where m is the number of periodic speed measurements.

The other input fuzzy variable is asscoaciated to the absolute value of vehicle acceleration. In congested streets vehicles' speed usually decreases and increases continuously; therefore the mean absolute

acceleration (MAA), which is the mean of the absolute value of vehicle's accelerations over a specific time interval, increases. MAA is computed as follows:

$$MAA_T = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{\Delta v_i}{\Delta t_i}\right| \qquad (2)$$

where, Δt_i is the interval of speed sampling, and n is the number of samples.

The Mamdani fuzzy inference system [34] is exploited to infer the traffic level. Three fuzzy sets of slow, medium, and fast are used for the speed variable, and three fuzzy sets of low, medium, and high are exploited for MAA variable. These membership functions are illustrated in Figure 2. On the other hand, four fuzzy sets are used for traffic level which include free flow, low congestion, medium congestion, and high congestion, as shown in Figure 2. Table 1 shows the proposed rule base.

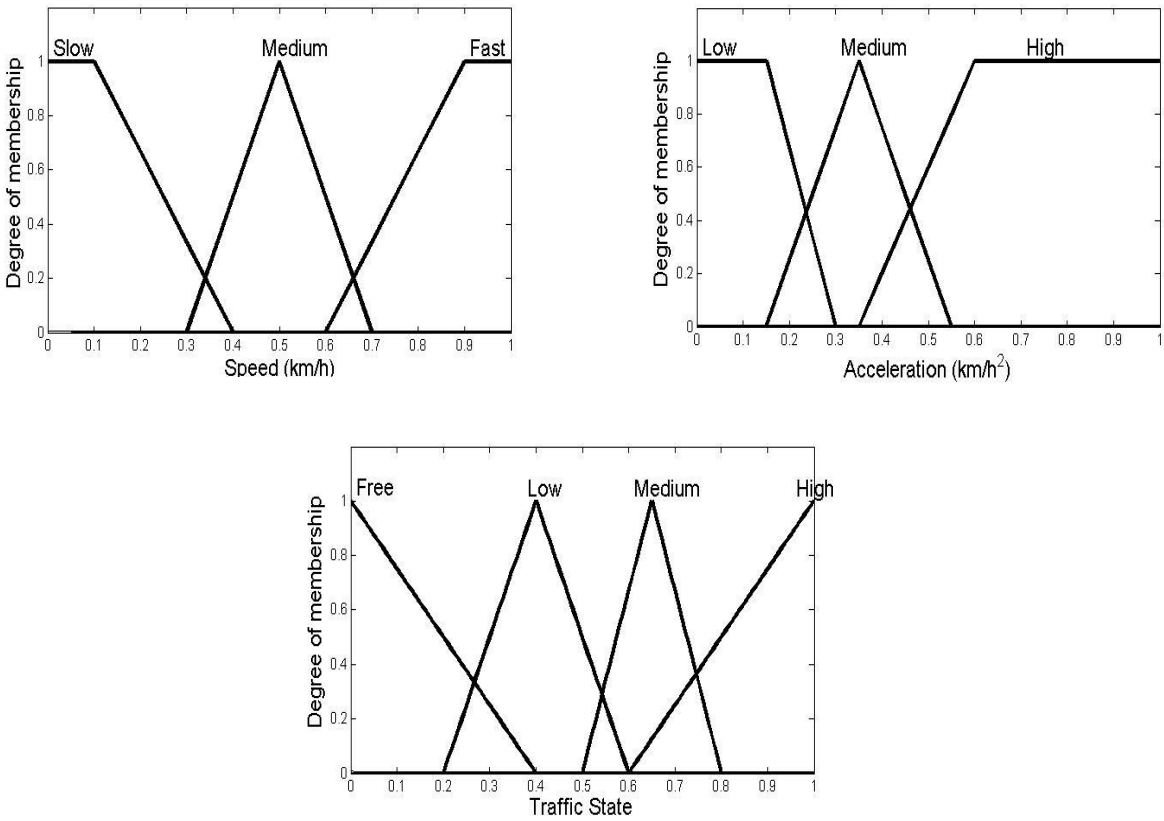

Figure 2: The membership functions used for fuzzy variables

Table 1: The fuzzy rule base of CANS

	Traffic State	**Acceleration**		
		Low	**Medium**	**High**
Speed	**Slow**	Medium	Medium	High
	Medium	Low	Low	Medium
	Fast	Free	Free	Low

For evaluating and aggregating the rules, the Max-Min [35] inference is exploited, which is formulated as bellow:

$$\mu_F(U) = max\{min_{i=1}^{N} [\alpha_i , \mu_{F_i}(U)] \} \qquad (3)$$

,Where N is the number of rules and μ_F (U) is the membership function of the fuzzy set F. α_i is the fuzzy degree of the i-th rule with n inputs x_1, x_2,…, x_n that is determined as follows:

$$\alpha_i = min[\mu_{A_1}(x_1), \mu_{A_2}(x_2) \dots \mu_{A_n}(x_n)] \qquad (4)$$

Finally, the acquired output fuzzy set is transformed to a crisp value via defuzzification. For this, center of gravity (COG) [34] is exploited, which catches the geometrical center of the fuzzy variable and is obtained as follows:

$$x_{COG} = \frac{\sum_{i=1}^{N} x_i \mu_f(x_i)}{\sum_{i=1}^{N} \mu_f (x_i)} \qquad (5)$$

,Where x_i is an element in the output fuzzy set, μ_f (x) is fuzzy set membership function, and N is the number of fuzzy rules. The crisp output of the fuzzy system is a number between 0 and 1, which determines the traffic level. A value near to zero or one shows a traffic-free or traffic jam situation, respectively.

3-1-2 Global Traffic Estimation

Any vehicle in CANS computes its local street traffic level at any time period. The local traffic level measured by different vehicles on different sections of a street segment may have different values so that the local traffic level measured by a vehicle cannot reflect the global traffic state of the whole street. Therefore, a model is required that can aggregate and estimate the global traffic state of the street. For this purpose, a global traffic estimation component is considered in CANS. As it can be seen in Figure 1, this component is located on the urban server. Any vehicle computes its local traffic level at any certain time period, creates a traffic sample in the form of {LinkID, Traffic}, and sends it to the urban server periodically. *LinkID* is a unique value for each street segment on the whole road network; and *Traffic* is the local traffic level estimated by the vehicle. Any vehicle is equipped with V2I communication technology to transmit the locally computed traffic information to the urban server. Thus, the urban server receives different traffic samples from the vehicles for every street segment. Its major role is to aggregate these pieces of various traffic information. The scheme, which is called crowd sensing is depicted in Figure 3.

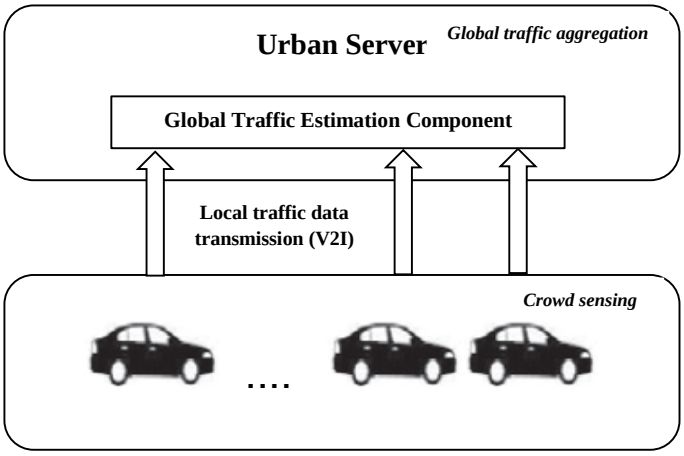

Figure 3: Global traffic aggregation in the CANS

The global traffic estimation component on the urban server updates the street's global traffic information upon receiving each traffic sample, and stores the updated information on the global traffic information database. The global traffic estimation component updates and aggregates a street global traffic state by the following weighted formula:

$$Traffic_{Global} = 0.9\, Traffic_{Old} + 0.1\, Traffic_{New} \qquad (6)$$

Traffic $_{New}$ is the received traffic sample from a vehicle; *Traffic* $_{Old}$ is the previous traffic level of the street received from the global traffic information database; and *Traffic* $_{Global}$ is the new, aggregated traffic value of the street. The weights of the input paprameters are adjusted by try and error.

3-2 Traffic-aware Navigation

The second functionality of CANS is traffic-aware navigation, which helps vehicles to find their optimal route to reach the specified destination. For this purpose, a navigation component is devised on the urban server. This component finds the current fastest route between the starting point and destination based on the current traffic information of routes. The driver demands for the optimal route, specifies the desired destination. The current location of the vehicle (the origin) is retrievable using a GPS receiver or other positioning systems [36-38]. Then, information about the two points is transmitted to the urban server. The navigation component on the urban server computes the fastest route between the specified origin and destination based on the current traffic state. The obtained route is offered to the driver along with the travel time.

In order to find the fastest route, the navigation component is required to estimate the travel time for each of the different routes from the starting point to the destination. Estimation of travel time on each street segment is a fundamental issue which depends on various parameters such as street length, traffic state, and speed limit. In the proposed system, travel time on each street link is computed by the following formula:

$$Travel\ Time_{Link}(t) = \frac{Link\ Length}{Speed\ Limit\ (1 - Traffic(t))} \qquad (7)$$

,in which *Link Length* is the length of street link, *Speed Limit* represents the maximum speed limit, and *Traffic (t)* indicates global traffic level at the time *t*.

To perform a dynamic estimation of travel time on each street segment, the navigation component receives recent traffic information from the global traffic information database. Static information such as street length and speed limit are accessible from the digital map. Then, the travel time for each street segment is computed by Formula 7.

The optimal route is considered as the fastest one. Therefore, the cost function is defined proportionate to the travel time. On the other hand, some intersections have traffic lights, and the waiting time at traffic lights is effective in travel time. In addition, crossing a square or an intersection without a traffic light also takes some extra time (delay). Thus, it is necessary to take into account both the travel time from the street segment and the delayed time at intersections or squares, including waiting time at traffic lights when the light is red. Nonetheless, mean waiting time at traffic lights and the delayed time at intersections or squares differ, suggesting that the delayed time should be computed for different intersections. Hence, the cost function for the weight of each street link is defined as follows:

$$Cost\ Function = Travel\ Time_{Link} + Delay \qquad (8)$$

,where *Travel Time* $_{Link}$ refers to the travel time on the street and *Delay* indicates the time delayed at intersections and squares. We use a statistical computation to estimate the expected delay time at intersections with traffic light. The delay time is modeled as a uniform random variable in the range [0 , *Max-Red-Time*], where *Max-Red-Time* indicates the maximum time of red state of the traffic light. Therefore, the expected delay time is computed as half of *Max-Red-Time*. For unsignalized intersections as well as squares, we perform an experiment to estimate the delay time at the experiments section.

The navigation problem is modeled as the search for the shortest path on a weighted graph. In this model, street segments are graph edges, intersections are graph nodes, and the weight of each edge is computed by the cost function in Formula 8. The navigation component uses Dijkstra algorithm [39] to find the shortest path between starting point and destination.

In general, the stages of the navigation algorithm are as follows:

Step 1: Corresponding nodes to the starting and destination points on the weighted graph representing city map are found.
Step 2: Dijkstra algorithm is performed, and the shortest path to reach the destination is found in terms of time.
Step 3: The path is displayed to the user along with relevant information including travel time and length.

4- Experiments

SUMO [40] simulator is exploited to simulate vehicular mobility. The evaluation is performed on an urban network scenario. For this aim, a map of a region of Birjand city, which is the capital of a province in eastern Iran, is exported from OpenStreetMap [41]. The region has an area of 7 km * 6 km. It includes 362 street segments and 102 intersections. Figure 4 shows the map of the region. It consists of low, medium and high traffic congested streets in order to cover enough varieties to be acceptable for the experiment. In this road network, the speed limit of the major streets is considered 60 km/h and 40 km/h for auxiliary streets (according to driving policy). A summary of the simulation parameters used in the scenario are shown in Table 2.

Before proceeding, an analysis of the network cost function of the CANS is provided. All vehicles perform V2I communications with the urban server to aggregate the traffic state. Each message consists of a real number indicating local traffic state (4 Bytes), and an integer indicating Link ID (2 Bytes). Therefore, each vehicle totally sends 6 Bytes in each 25 seconds interval. We assume that at a moment in a metropolis, 100,000 vehicles are driving and communicating with the server. Hence, the consumed bandwidth of the aggregation server is 24 KB/s, which is negligible.

In the following, the scenario is evaluated by the traffic flow of Birjand. For this purpose, the urban network of Birjand as well as streets' traffic information are imported into SUMO.

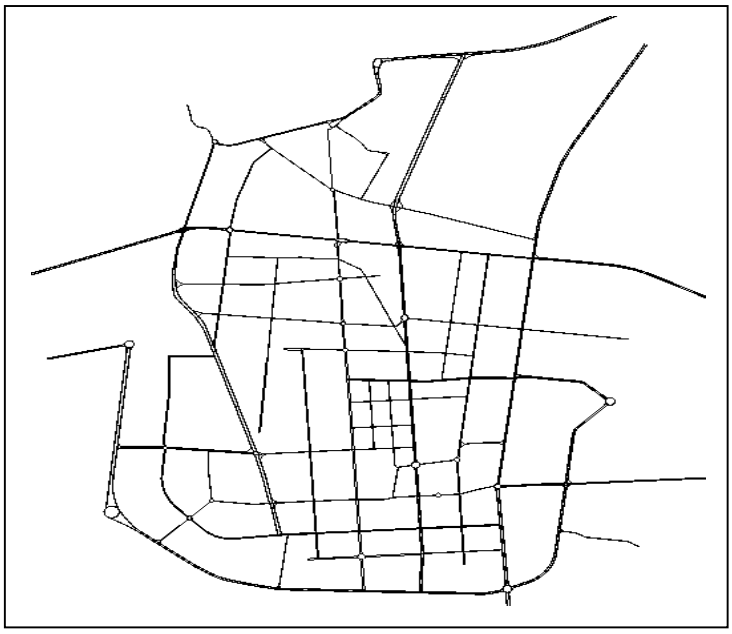

Figure 4: The road network of Birjand

Table 2: The parameters used in the simulation scenario

Parameter	Value
Simulation area	Birjand - 7 km * 6 km
Speed limit	40 , 60 km/h
Number of links	362
Number of intersections	102
Vehicle maximum speed	120 km/h
Vehicle acceleration rate	5.4 km/h^2
Vehicle deceleration rate	16.2 km/h^2

4-1 Traffic Estimation Validation

Every vehicle on each street segment gathers its current speed periodically at 5 second intervals. Then, it calculates the average speed and mean absolute acceleration at 25 second time period using the recorded instantaneous speeds. These numbers are selected by consultations of the urban traffic experts as well as try and error. Finally, the local traffic level is estimated and transmited it to the urban server.

Traffic measurement is performed with the penetration rate of 10%; that means, only 10% of the total number of vehicles transmit the local traffic value to the urban server. On each street segment, vehicles are sampled randomly from different locations. In SUMO, we derived the local traffic level values

computed by the sample vehicles over the simulation time. Building on local traffic levels estimated by sample vehicles, the urban server computes the global traffic congestion level of the streets. These values are displayed in the second column of Table 3 for some major streets.

Mean as well as standard deviation of local traffic level obtained by some randomly generated vehicles that are located on the same street, are computed and displayed in Table 3. Low values of standard deviation suggest that the available vehicles on the same street have computed similar traffic levels, validating the logic of the proposed fuzzy system. In addition, the comparison between the mean value of traffic congestion levels and the global traffic level aggregated and measured by the urban server reveals that the amount of traffic is relatively stable during a short time period, despite various traffic samples are made by different vehicles on different sections of a street segment.

Table 3: Global traffic congestion level validation

Street Name	Global Traffic Congestion Level	Mean	Standard Deviation
Modares Blvd.	0.64672	0.6533	0.00114
Moallem Blvd.	0.5364	0.5546	0.07725
Pasdaran Blvd.	0.14782	0.1453	0.00662
Taleghani St.	0.65215	0.6533	0.00004
Beheshti Blvd.	0.47231	0.4618	0.07222
Motahari Blvd.	0.4037	0.4067	0.00445
Mohalati Blvd.	0.30549	0.3328	0.13231
Emamat St.	0.18252	0.1847	0.03168

Since the proposed system is based on estimating travel time, we compare the accuracy of travel time estimated by CANS and CATE [29] on a number of major streets. The benchmark for comparison is the travel time computed by SUMO simulator, i.e., average travel time observed by 10 vehicles in SUMO. The results obtained from the evaluation of travel time are displayed in Table 4. As it can be seen, the proposed system has estimated a better approximation on 12 streets, while CATE has shown a better approximation only on 3 streets.

For better comparison, absolute differential diagram of travel time estimated by CANS and CATE and the travel time measured by SUMO simulator on these streets are depicted in Figure 5. Results indicate that on the majority of these streets the travel time estimated by the proposed approach is closer to that of SUMO, and its travel time differential is less. This suggests the higher accuracy of the proposed system than CATE in the simulated urban scenario.

Table 4: Comparison of the travel time estimated by CANS and CATE with respect to the travel time measured by SUMO

Street Name	Travel Time Estimated by CANS (seconds)	Travel Time Estimated by CATE (seconds)	Travel Time Measured by SUMO (seconds)
Beheshti Blvd.	187.04	160	192
Motahari Blvd.	135.25	98	127
Montazeri St.	159.41	122	163
Shohada St.	57.912	60	59
Taleghani St.	215.88	237	214
Modares Blvd.	205.29	156	192
Pasdaran Blvd.	231.09	228	230

Moallem Blvd.	595.19	550	584
Ghafari Blvd.	376.56	318	380
Artesh St.	164.53	167	170
Pyambr Azam Blvd.	385.81	373	378
Emamat St.	305.21	289	302
15khordad St.	244.15	261	251
Tohid St.	223	218	225
Mohalati Blvd.	223.41	265	225

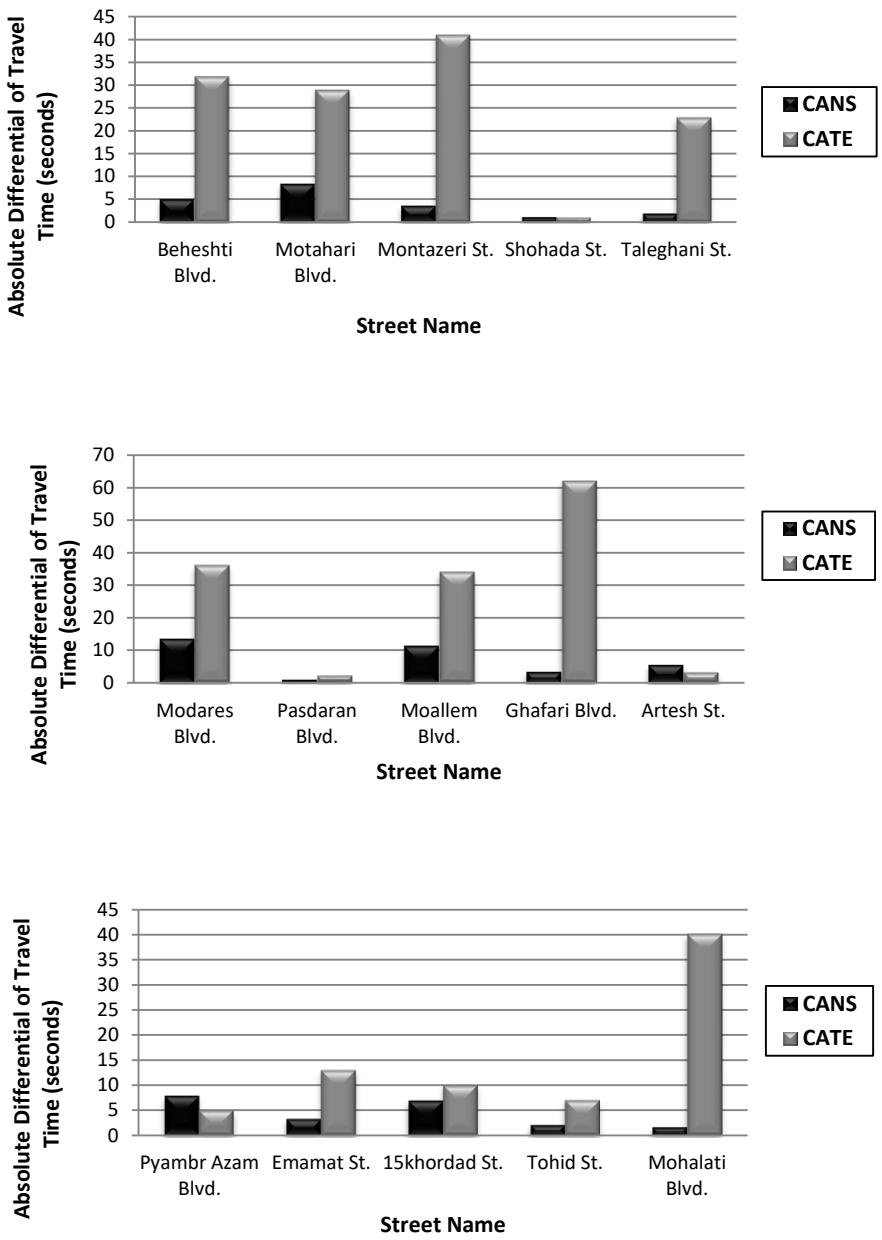

Figure 5: Absolute differential of the estimated travel times and the simulated travel time by SUMO

4-2 Navigation Evaluation

The proposed navigation system is compared with CATE dynamic navigation system as well as static navigation system. For this purpose, five different origin-destination pairs have been selected semi-randomly. The optimal routes between the origins and destinations have been computed by CANS, CATE, and static navigation system. The results are displayed in Table 5.

Static navigation finds the shortest route between the specified origin and destination in terms of distance. CANS uses the cost function given in Formula 8 in order to compute the weight of each street segment. For this purpose, an experiment is conducted to empirically calculate the mean delay of unsignalized intersections and squares. To this end, the delay time is empirically computed and averaged for 20 unsignalized intersections and squares. It has been computed as the difference between crossing the intersection (or square) without decreasing the speed and the normal crossing. As a result, the mean delay time of 5 seconds has been estimated as the average of these experiments for crossing a square or unsignalized intersection. It seems a reasonable time, because unsignalized intersections and squares have low traffic congestion level; Otherwise, they should logically be signalized. The mean waiting time at traffic lights is assumed half of the maximum red-light time. Traffic lights are divided into short-term and long-term categories. The mean waiting time at short-term and long-term traffic lights is computed as 15 and 30 seconds, respectively.

Travel time of the routes estimated by CANS, CATE, and static navigation system for the major origin-destination pairs are compared in Figure 6. As it can be observed, CANS system considerably reduces the travel time for the majority of routes comparing with the static navigation. The main reason for that is the integration of streets' traffic information. CATE and CANS have reported the same routes for three but different routes for two cases. In both situations, the travel time of the route suggested by CANS is less than that of CATE.

Table 5: Navigation routes obtained from the static navigation, CANS and CATE

Origin- Destination	Approach	Estimated Route (series of road link names)	Route Time (seconds)	Route Distance (meter)
Taleghani Sq.- Third Modares Sq.	Static Navigation	(Taleghani St.- Modares Blvd.)	522.4	1911.94
	CANS	(Beheshti Blvd.- Navab Safavi St.- Modares Blvd.)	339	2051.7
	CATE	(Beheshti Blvd.- Navab Safavi St.- Modares Blvd.)	339	2051.7
Velayat Sq.- End of the South Emamat St.	Static Navigation	(Ghadir Blvd.- Sajjad Blvd.)	160	1500.87
	CANS	(Ghadir Blvd.- Sajjad Blvd.)	160	1500.87
	CATE	(North Emamat St.- South Emamat St.)	175.7	1559.35
Jamaran Sq.- Abouzar Sq.	Static Navigation	(Modares Blvd.)	505.9	2862.62
	CANS	(Jamaran Blvd.- Pasdaran Blvd.- Artesh St.)	378.7	3847.21
	CATE	(Jamaran Blvd.- Pasdaran Blvd.- Artesh St.)	378.7	3847.21
	Static Navigation	(Ghafari Blvd.- Ghods Gharbi St.- Navab Safavi St.- Modares Blvd.)	346.8	2349.72

Beginning of the Ghafari Blvd.- Third Modares Sq.	CANS	(Artesh St.- Pasdaran Blvd.- Bahonar Gharbi St.)	272.2	2529.72
	CATE	(Ghafari Blvd.- Ghods Gharbi St.- Pasdaran Blvd.- Bahonar Gharbi St.)	298	2381.9
Intersection between Moallem Blvd. and Ghods Gharbi St.- 7Tir Sq.	Static Navigation	(Moallem Blvd.- 15khordad St.)	365	1832.35
	CANS	(Ghods Gharbi St.- Pasdaran Blvd.- 15khordad St.)	233.12	2494.17
	CATE	(Ghods Gharbi St.- Pasdaran Blvd.- 15khordad St.)	233.12	2494.17

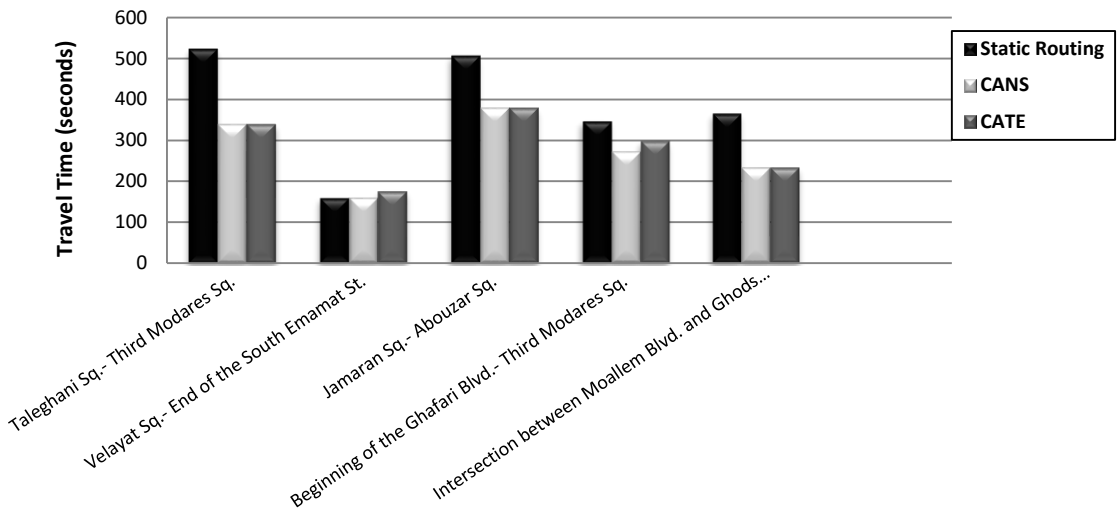

Figure 6: Travel time of estimated routes for 5 major origin-destination pairs

5- Conclusion

A context-aware system for dynamic traffic estimation and traffic-aware vehicular navigation has been proposed. The traffic state is estimated by vehicles without any requirement for any specific costly infrastructure. As mobile traffic sensors, vehicles build on fuzzy logic to estimate their current street traffic state using local contextual information. Later, the urban server integrates the pieces of local traffic information measured by vehicles to estimate road global traffic state, offering a traffic-aware navigation system. The system has been evaluated on an urban network scenario with realistic traffic flow. The experimental results confirm that concurrent use of vehicle's local contextual information for traffic state identification has estimated acceptable traffic state in various traffic situations. In addition, the proposed navigation system has been compared with static and CATE dynamic navigation systems. Navigation results indicate the efficacy of the proposed system for urban networks navigation such that it could reduce travel time in some of routes more than CATE and static navigation. Future research directions are as follows:

- In the current study, a few factors were used for traffic estimation. Other influential traffic factors can be incorporated to further the accuracy of the proposed system such as road slope, weather conditions, etc.

- In this research, upon receiving the user's navigation request, navigation is performed once according to the newest traffic information. Any new navigation should be initialized by the user. Real-time re-navigation as a result of changes in traffic information is a big issue that is considered as the future direction of this paper.

- Dissemination and exchange of real-time traffic information between vehicles and road-side infrastructure is a fundamental challenge on vehicular networks. It requires the design and development of intelligent routing protocols for establishing effective communication and improving information dissemination mechanism between vehicles.

- The estimated traffic state can be utilized to develop useful traffic applications in the transportation network such as traffic management at intersections without traffic lights, scheduling traffic lights based on dynamic traffic information, etc.

References

[1] F. Li, and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *Vehicular Technology Magazine, IEEE,* vol. 2, no. 2, pp. 12-22, 2007.

[2] H. Hartenstein, and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *Communications Magazine, IEEE,* vol. 46, no. 6, pp. 164-171, 2008.

[3] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Computer communications, Elsevier,* vol. 31, no. 12, pp. 2838-2849, 2008.

[4] J. Rybicki, B. Scheuermann, W. Kiess, C. Lochert, P. Fallahi, and M. Mauve, "Challenge: peers on wheels-a road to new traffic information systems," in Proceedings of the 13th annual ACM international conference on Mobile computing and networking (MobiCom), Montreal, Canada, 2007, pp. 215-221.

[5] J. Liu, J. Wan, Q. Wang, P. Deng, K. Zhou, and Y. Qiao, "A survey on position-based routing for vehicular ad hoc networks," *Telecommunication Systems,* vol. 62, no. 1, pp. 15-30, 2016.

[6] H. Moustafa, and Y. Zhang, *Vehicular networks: techniques, standards, and applications*: Auerbach publications, 2009.

[7] S. Olariu, and M. C. Weigle, *Vehicular networks: from theory to practice*: CRC Press, 2009.

[8] V. Jindal, and P. Bedi, "Vehicular Ad-Hoc Networks: Introduction, Standards, Routing Protocols and Challenges," *International Journal of Computer Science Issues (IJCSI),* vol. 13, no. 2, pp. 44-55, 2016.

[9] H. Vahdat-Nejad, A. Ramazani, T. Mohammadi, and W. Mansoor, "A survey on context-aware vehicular network applications," *Vehicular Communications, Elsevier,* vol. 3, pp. 43-57, 2016.

[10] Y. Cho, and J. Rice, "Estimating velocity fields on a freeway from low-resolution videos," *IEEE Transactions on Intelligent Transportation Systems,* vol. 7, no. 4, pp. 463-469, 2006.

[11] B. T. Morris, and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Transactions on Intelligent Transportation Systems,* vol. 9, no. 3, pp. 425-437, 2008.

[12] Y. Xia, X. Shi, G. Song, Q. Geng, and Y. Liu, "Towards improving quality of video-based vehicle counting method for traffic flow estimation," *Signal Processing, Elsevier,* vol. 120, pp. 672-681, 2016.

[13] B. Coifman, “Improved velocity estimation using single loop detectors,” *Transportation Research Part A: Policy and Practice, Elsevier,* vol. 35, no. 10, pp. 863-880, 2001.

[14] B. Coifman, S. Dhoorjaty, and Z.-H. Lee, “Estimating median velocity instead of mean velocity at single loop detectors,” *Transportation Research Part C: Emerging Technologies, Elsevier,* vol. 11, no. 3, pp. 211-222, 2003.

[15] C. C. Sun, G. S. Arr, R. P. Ramachandran, and S. G. Ritchie, “Vehicle reidentification using multidetector fusion,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 5, no. 3, pp. 155-164, 2004.

[16] W. L. Leow, D. Ni, and H. Pishro-Nik, “A sampling theorem approach to traffic sensor optimization,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 9, no. 2, pp. 369-374, 2008.

[17] B. R. Hellinga, and L. Fu, “Reducing bias in probe-based arterial link travel time estimates,” *Transportation Research Part C: Emerging Technologies,* vol. 10, no. 4, pp. 257-273, 2002.

[18] Y. Li, and M. McDonald, “Link travel time estimation using single GPS equipped probe vehicle,” in Proceedings 5th International IEEE Conference on Intelligent Transportation Systems, 2002, pp. 932-937.

[19] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, “A compressive sensing approach to urban traffic estimation with probe vehicles,” *IEEE Transactions on Mobile Computing,* vol. 12, no. 11, pp. 2289-2302, 2013.

[20] J. Wan, D. Zhang, S. Zhao, L. Yang, and J. Lloret, “Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions,” *IEEE Communications Magazine,* vol. 52, no. 8, pp. 106-113, 2014.

[21] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano, “Road traffic congestion detection through cooperative vehicle-to-vehicle communications,” in Proceedings 35th IEEE Conference on Local Computer Networks (LCN), 2010, pp. 606-612.

[22] X. Jiang, and D. H. Du, “BUS-VANET: a bus vehicular network integrated with traffic infrastructure,” *IEEE Intelligent Transportation Systems Magazine,* vol. 7, no. 2, pp. 47-57, 2015.

[23] Y. Huang, J. Wang, C. Jiang, H. Zhang, and V. C. Leung, “Vehicular Network Based Reliable Traffic Density Estimation,” in Vehicular Technology Conference (VTC Spring), IEEE, 2016.

[24] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," *Handheld and ubiquitous computing*, H.-W. Gellersen, ed., pp. 304-307: Springer Berlin Heidelberg, 1999.

[25] A. K. Dey, “Understanding and using context,” *Personal and ubiquitous computing, Springer-Verlag,* vol. 5, no. 1, pp. 4-7, 2001.

[26] A. Ramazani, and H. Vahdat-Nejad, “A new context-aware approach to traffic congestion estimation,” in 4th International eConference on Computer and Knowledge Engineering (ICCKE), IEEE, Mashhad, Iran, 2014, pp. 504-508.

[27] R. K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *IEEE Communications Magazine,* vol. 49, no. 11, pp. 32-39, 2011.

[28] B. Guo, Z. Yu, X. Zhou, and D. Zhang, “From participatory sensing to mobile crowd sensing,” in IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Budapest, 2014, pp. 593-598.

[29] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, and M. Gerla, “On the effectiveness of an opportunistic traffic management system for vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 12, no. 4, pp. 1537-1548, 2011.

[30] N. Zarei, M. A. Ghayour, and S. Hashemi, "Road traffic prediction using context-aware random forest based on volatility nature of traffic flows," *Intelligent Information and Database Systems*, A. Selamat, N. Thanh Nguyen and H. Haron, eds., pp. 196-205: Springer Berlin Heidelberg, 2013.

[31] P. Raphiphan, A. Zaslavsky, P. Prathombutr, and P. Meesad, “Context aware traffic congestion estimation to compensate intermittently available mobile sensors,” in Tenth International Conference on Mobile Data Management: Systems, Services and Middleware (MDM'09), IEEE, Taipei, 2009, pp. 405-410.

[32] N. Kim, H. S. Lee, K. J. Oh, and J. Y. Choi, “Context-aware mobile service for routing the fastest subway path,” *Expert Systems with Applications, Elsevier,* vol. 36, no. 2, pp. 3319-3326, 2009.

[33] Y. Wang, J. Jiang, and T. Mu, “Context-aware and energy-driven route optimization for fully electric vehicles via crowdsourcing,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 14, no. 3, pp. 1331-1345, 2013.

[34] J. H. Lilly, "Mamdani Fuzzy Systems," *Fuzzy Control and Identification*, pp. 27-45: John Wiley & Sons, 2010.

[35] I. Baturone, A. Barriga, C. Jimenez-Fernandez, D. R. Lopez, and S. Sanchez-Solano, *Microelectronic design of fuzzy logic-based systems*: CRC press, 2000.

[36] I. Sabek, M. Youssef, and A. V. Vasilakos, “ACE: An accurate and efficient multi-entity device-free WLAN localization system,” *IEEE Transactions on Mobile Computing,* vol. 14, no. 2, pp. 261-273, 2015.

[37] K. Subbu, C. Zhang, J. Luo, and A. Vasilakos, “Analysis and status quo of smartphone-based indoor localization systems,” *IEEE Wireless Communications,* vol. 21, no. 4, pp. 106-112, 2014.

[38] J. Liu, J. Wan, Q. Wang, B. Zeng, and S. Fang, “A time-recordable cross-layer communication protocol for the positioning of vehicular cyber-physical systems,” *Future Generation Computer Systems, Elsevier,* vol. 56, pp. 438-448, 2016.

[39] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik, Springer-Verlag,* vol. 1, no. 1, pp. 269-271, 1959.

[40] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO–simulation of urban mobility,” *International Journal On Advances in Systems and Measurements,* vol. 5, no. 3&4, pp. 128-138, 2012.

[41] "OpenStreetMap homepage [Online]," from:http://www.openstreetmap.org/.