This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Safe and Sound: Driver Safety-Aware Vehicle Re-Routing Based on Spatiotemporal Information

Allan M. de Souza, Torsten Braun, Leonardo C. Botega, Leandro A. Villas, and Antonio A. F. Loureiro

***Abstract*— Vehicular traffic re-routing is key to provide better vehicular mobility. However, considering just traffic-related information to recommend better routes for each vehicle is far from achieving the desired requirements of a good Traffic Management System, which intends to improve not only mobility but also driving experience and safety of drivers and passengers. Context-aware and multi-objective re-routing approaches will play an important role in traffic management. However, most of these approaches are deterministic and can not support the strict requirements of traffic management applications, since many vehicles potentially will take the same route, and, thus, degrade the overall traffic efficiency. In this work, we introduce Safe and Sound (SNS), a non-deterministic multi-objective re-routing approach for improving traffic efficiency and reduce public safety risks (based on criminal events) for drivers and passengers. SNS employs a hybrid architecture and a cooperative re-routing approach for improving system scalability and computation efforts. SNS uses a recurrent neural network to both predict future safety risks dynamics and enable a personalized re-routing in which each vehicle decides the risks it wants to avoid. Simulation results revealed that when compared to state-of-the-art approaches, SNS reduces the CPU time of the re-routing algorithm in approximately 99% and decreases the average safety risk for drivers and passengers in at least 30% while keeping efficient traffic mobility.**

***Index Terms*— Vehicle routing, vehicle safety, advanced driver assistance.**

I. Introduction

RECENT studies have shown that vehicular traffic re-routing is key to improve traffic efficiency [14], [30]. However, considering just traffic-related information to recommend better routes for each vehicle is far from achieving the desired requirements of a Traffic Management System (TMS). In this sense, context-awareness and multi-objective re-routing will play an essential role in vehicular traffic re-routing [2], [37], paving the way for improving the vehicular experience and enabling a whole new set of services to enhance not only traffic mobility but also driving experience, energy consumption of electric vehicles as well as the safety of drivers and passengers [11], [17].

Context-awareness in vehicular re-routing is essential since different drivers can have different preferences to perform their journey [2]. These preferences are related to the urban, drivers and vehicle contexts such as travel time, distance, fuel consumption, scenery and certainly safety, which can lead to different routes to reach the same location [37]. However, just considering single preferences to re-route vehicles can create other significant concerns. For instance, a couple got shot and the woman died after taking the directions recommended by a vehicular navigation system (VNS), which guided them toward a dangerous neighborhood in Rio de Janeiro, Brazil [9]. Another example shows a vehicle (which took the directions recommended by a VNS) passing through a shooting in Boston [29]. These issues could have been avoided using safe route recommendation systems [20], [35]. Navigation systems that focus on optimal safety can also lead to stressful paths, since the recommendation algorithm might include congested roads to provide the safest way, and these roads are more likely to be avoided in drivers' criteria during their route planning [2]. In this scenario, a multi-objective optimization of traffic efficiency and safety is desirable to increase the appeal and the effectiveness of the re-routing strategy.

Advances in wireless communication and processing such as the fifth generation (5G) mobile networks [18], vehicle-to-everything (V2X) communication [6] and multi-access edge computing (MEC) [24] will enable TMSs to sense and act in the urban environment in different ways, interacting not only with vehicles, but also with smart devices, subsystems and even people, in order to provide better solutions [22]. In other words, TMSs will be able to understand a set of different urban factors to extract various pieces of knowledge, which will help in their traffic management decisions including detecting areas with recurrent traffic congestion and limited safety, improve re-routing effectiveness, and avoid dangerous neighborhoods. Besides, with the help of machine learning techniques [41], TMSs can predict future urban dynamics and know in advance when a given area can become congested or dangerous to improve their effectiveness. However, how to predict these urban dynamics accurately and explore their spatiotemporal correlation is still an open issue. In this scenario, deep learning techniques such as recurrent neural networks (RNN) [25], [28],

Manuscript received May 2, 2019; revised September 19, 2019; accepted November 18, 2019. This work was supported by São Paulo Research Foundation (FAPESP) under Grant #2018/19639-5 and Grant #2019/24937-8. The Associate Editor for this article was J. Sanchez-Medina. *(Corresponding author: Allan M. de Souza.)*

A. M. de Souza is with the Institute of Computing, University of Campinas, Campinas 13083-970, Brazil, and also with the Institute of Computer Science and Applied Mathematics, University of Bern, 3012 Bern, Switzerland (e-mail: allanms@lrc.ic.unicamp).

T. Braun is with the Institute of Computer Science and Applied Mathematics, University of Bern, 3012 Bern, Switzerland (e-mail: torsten.braun@inf.unibe.ch).

L. C. Botega is with the Information Science Department, State University of São Paulo, São Paulo 01049-010, Brazil (e-mail: leonardo.botega@unesp.br).

L. A. Villas is with the Institute of Computing, University of Campinas, Campinas 13083-970, Brazil (e-mail: leandro@ic.unicamp.br).

A. A. F. Loureiro is with Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte 31270-901, Brazil (e-mail: loureiro@dcc.ufmg.br).

Digital Object Identifier 10.1109/TITS.2019.2958624

[33], [40] can play an important role, by providing accurate predictions about urban dynamics such as traffic conditions and safety risks, while exploring their spatial and temporal information.

Several solutions have been proposed to enable context-awareness and multi-objective re-routing [26] such as Weighted-Sum [27], Resource Constrained Shortest Path (RCSP) [21] and Evolutionary algorithms [23]. However, as we have shown in our previous work [10], most of these solutions are deterministic and are not suitable for traffic management applications, since many vehicles with the same origin and destination can take the same route, potentially degrading traffic efficiency. Thus, we proposed EBPOP (Entropy Balanced Pareto-optimal Path) a non-deterministic context-aware multi-objective vehicular re-routing algorithm, which overcomes the limitations presented in the literature considering a fast and safe path use case [10].

EBPOP aims to balance the traffic flow over the routes that compose the Pareto-set for a particular origin and destination pair, e.g., the set of paths that optimize both traffic efficiency and safety. The knowledge about traffic conditions is obtained by traffic reports sent by all vehicles, while the awareness about safety risks is extracted based on the criminal events over the city. Despite its better performance against state-of-the-art approaches, EBPOP still presents some limitations related to (*i*) complexity time of the re-routing algorithm and system scalability; (*ii*) spatiotemporal correlation of each type of criminal activity (e.g., robbery, assault, narcotics, weapons violation, and kidnapping) and its safety risk; (*iii*) lack of knowledge about future safety risks in advance (e.g., it does not provide any prediction about safety risks); and (*iv*) it does not enable drivers to choose which type of criminal activities (e.g., safety risks) they want to avoid.

Motivated by the limitations mentioned above, in this paper, we propose **S**afe a**n**d **S**ound (SNS), which is an extension of EBPOP [10] that overcomes the limitations previously introduced. Therefore, the main contributions of this work are as follows:

Cooperative context-aware re-routing system: The proposed non-deterministic context-aware multi-objective re-routing algorithm presented in EBPOP [10] has complexity of $O(|E|\lambda)$, in which $|E|$ is the number of roads in the scenario and λ is a value representing the current safety risk of the vehicle's route (more details are presented in Section VI). Moreover, EBPOP employs a centralized architecture to provide vehicular traffic re-routing, which increases the complexity of the algorithm by the number of vehicles to be re-routed. Therefore, SNS uses a distributed approach to overcome such a problem. This strategy offloads the re-routing procedure in each vehicle. Consequently, it is not affected by the number of vehicles to be re-routed. However, a cooperative approach needs to be implemented into the algorithm to enable the same performance of the centralized approach. Also, to improve the system scalability, a traffic reporting protocol is introduced to reduce network overhead.

Prediction of future safety risks: Knowing in advance the future safety risks of relevant areas of the city is crucial for improving the safety of drivers and passengers during re-routing. Moreover, criminal activities have a spatiotemporal correlation, in which the safety risk of some neighborhood can be different along the day – for instance, it can either increase or decrease for specific types of crimes in such area. In this way, we propose an RNN to predict future safety risks of each crime along the day considering the past events in each area and exploiting the spatiotemporal correlation. To evaluate this prediction, we used the official criminal records dataset of Chicago.

Personalized safety risk avoidance: Different drivers and passengers can have different preferences related to the safety risks provided by each criminal activity. Thus, some drivers are more likely to avoid a specific type of crime than others. Therefore, it is essential to let the drivers themselves (e.g., vehicles) to decide which type of crime they want to avoid. In this context, as SNS offloads the re-routing algorithm in each vehicle, it enables such personalized feature. Thus, during re-routing, each vehicle can decide the safety risk priority on-the-fly, e.g., which criminal event provides higher safety risks for them.

Spatiotemporal analysis based on safety risks: The safety risks of each neighborhood can change depending on the day (e.g., business days and weekends) and time (e.g., morning, afternoon and night) for each criminal activity. Therefore, we analyze how the predictions of future safety risk dynamics and the personalized re-routing algorithm can work together to improve the safety of drivers and passengers considering the spatial and temporal characteristics of each criminal activity.

The paper is organized as follows. Section II discusses the related work highlighting their advantages and limitations. Section III presents the system overview. Section IV describes the traffic reporting protocol implemented by SNS and its traffic estimation. Section V presents safety risk estimations and the RNN implemented to predict future safety risks considering the spatiotemporal information. Section VI describes the cooperative context-aware re-routing algorithm and how it enables the personalized re-routing. Section VII analyzes the performance of the SNS compared to state-of-the-art solutions. Finally, Section VIII presents conclusions and future work.

II. Related Work

This section presents the related work highlighting aspects related to traffic re-routing, safe routing recommendation and multi-objective routing optimization approaches.

Services such as INRIX[1] provide real-time traffic information, which might help drivers to choose routes as fast as possible. In turn, Google Maps and Waze can forecast traffic congestion and its duration by performing advanced statistical predictive analysis of traffic patterns. Moreover, they also can recommend the fastest routes whenever route planning is requested.

Another approach for providing better routing guidance is enabled by an Intelligent Transportation System (ITS), which focuses on building pieces of knowledge about city-wide traffic conditions based on traffic-related measures provided by vehicles, in-road and roadside infrastructures [17]. In this

[1] http://www.inrix.com

way, based on that city-wide traffic view, dynamic re-routing approaches are employed to improve the overall traffic efficiency. For instance, Doolan *et al.* [19] proposed EcoTrec, an eco-friendly ITS, which intends to reduce carbon emissions and improve traffic efficiency based on a global traffic view built based on information reported by all vehicles in the network about fuel consumption and road characteristics. In addition, Cao *et al.* [8] combined a vehicular re-routing solution with a traffic light control to reduce traffic congestion and improve traffic efficiency.

On the other hand, motivated by high criminal activities over the city, safety-based solutions have been proposed [20], [35]. The key idea is to recommend the safest routes for all vehicles based on pieces of knowledge about dangerous areas and neighborhoods built from criminal incidents reports, which can be obtained either from official criminal statistics [20] or from social media and participatory sensing systems [35]. In this sense, Shah et. al proposed CrowdSafe [35] a crowdsourcing-based system that suggests safe routes for the drivers and also allows people to report crime-related incidents.

All aforementioned approaches, however, share the same problem: (*i*) they are all based on deterministic algorithms for recommending either the fastest or the safest route (e.g., the optimal) to each vehicle, and, consequently, create bottlenecks in the transportation infrastructure, since the same route can be recommended for many vehicles; and (*ii*) they can operate only with a single metric (mobility or safety), but whenever possible both criteria should be considered simultaneously.

The highlighted problems can be solved using multi-objective optimization algorithms [26] and by distributing the route recommendation over a set of feasible routes [15], [30]. Unfortunately, using purely multi-objective optimization approaches such as Resource Constrained Shortest Path [21], Weighted-Sum [27] and Evolutionary algorithms [23] can potentially decrease traffic efficiency, since they are also deterministic.

Souza *et al.* [15] introduced SCORPION, a cooperative re-routing system to prevent and improve the traffic efficiency. SCORPION is a centralized system that balances the traffic flow over a set of k-shortest paths computed based on current travel time on the roads. Traffic balancing is cooperative, which means that the system focuses on improving the traffic in the whole scenario rather than optimizing the route of each vehicle (i.e., greedy decisions). The cooperative approach contributes to better traffic management. However, it still does not support multiple metrics to re-route vehicles, leading to the same problems as [14], [16], [19], [30], [35].

Cao *et al.* [7] proposed a multiagent-based approach for vehicle re-routing, which considers the time to arrive at a destination and the total travel time as re-routing metrics. The proposed system is semi-decentralized composed of vehicles and infrastructures. In the proposed system, vehicle agents follow route recommendations provided by infrastructure agents deployed at each intersection, which recommend a route by solving a route assignment problem. The arrival time (i.e., the deadline) is modeled based on the tail probability to maximize the probability of reaching the destination in the desired time. On the other hand, the total travel time is formulated as a weighted quadratic term to minimize the expected travel time. In addition, a travel time prediction is used to improve the performance of the proposed solution. In this way, the proposed approach could be modeled to improve mobility (i.e., total travel time) and safety (i.e., maximum acceptable safety risk instead of time to arrive). However, the need of infrastructure at each intersection of the scenario potentially turns this approach into a costly solution.

In our previous work, we proposed EBPOP [10], an efficient context-aware vehicular traffic re-routing system based on Pareto-efficiency. The key idea is to find the set of routes that composes the Pareto set (e.g., the set of routes that optimally improves the traffic efficiency and safety risk given an origin and destination). By using an entropy-based approach, we can distribute the vehicles to be re-routed over the set of feasible routes, and, consequently, avoid the problem of creating additional congestion spots and still be able to handle mobility and safety risk issues. However, EBPOP has some problems not only related to time complexity and scalability due to its architecture but also related to the limited knowledge about future safety risks and their spatiotemporal correlation. Moreover, EBPOP does not enable personalized re-routing which is an essential feature for an efficient TMS, since different drivers can have different preferences according to the type of crime (e.g., safety risks) that they want to avoid.

Table I summarizes proposals discussed here to improve traffic efficiency and safety risks, highlighting their advances and limitations related to architecture, re-routing strategy, optimization, and re-routing considerations.

To tackle the discussed limitations, we propose SNS, a non-deterministic personalized context-aware re-routing system that incorporates safety risk prediction considering spatiotemporal information. SNS is described below.

III. SNS System Overview

The Safe and Sound (SNS) system is built around three design principles corresponding to the major requirements for improving vehicular traffic mobility and public safety. First, the communication mechanism needs to be optimized to reduce network overhead, and the re-routing algorithm must be offloaded from the central server to vehicles to reduce computation efforts and achieve better scalability. Second, vehicles need to be able to choose which safety risks they prefer to avoid (e.g., personalized risk selection). Third, the system must be able to predict future risky areas to know in advance the most dangerous regions for each safety risk that can be chosen by the vehicles.

Figure 1 shows the system architecture based on the design principles mentioned above. The architecture is composed of vehicles, edge servers, criminal-related data providers and cloud. Vehicles with on-board units (OBUs) can communicate with the roadside infrastructures (e.g., RSUs and 5G base station), edge servers and with the remote cloud (e.g., cellular communication and vehicular networking) to report traffic-related data, request/receive urban-related dynamics

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I

SUMMARY OF LITERATURE SOLUTIONS HIGHLIGHTING THEIR CHARACTERISTICS AND LIMITATIONS

	System design		Network Challenges		Routing strategy		Optimization		Re-routing considerations				
Related work	Centralized	Hybrid	Broadcast storm	Network contentions	Deterministic	Non-deterministic	Single-objective	Multi-objective	Traffic flow	Traffic light	Safety issues	Future dynamics	Drivers' preference
VNS	✓				✓		✓		✓				
DeepRSI [32]	✓					✓		✓	✓		✓		
DIVERT [30]		✓	✓	✓		✓	✓		✓				
Cao et al. [8]					✓			✓	✓	✓			
SCORPION [15]	✓					✓	✓		✓				
EcoTrec [19]		✓			✓		✓		✓				
NRR [39]		✓				✓	✓		✓				
ICARUS [14]		✓	✓	✓	✓		✓		✓				
SafePaths [20]	✓					✓		✓	✓		✓		
Crowdsafe [35]	✓				✓								
itsSAFE [12]	✓					✓		✓	✓		✓		
Cao et al. [7]		✓			✓			✓	✓			✓	
EBPOP [10]	✓					✓		✓	✓		✓		
SNS		✓	✓	✓		✓		✓	✓		✓	✓	✓

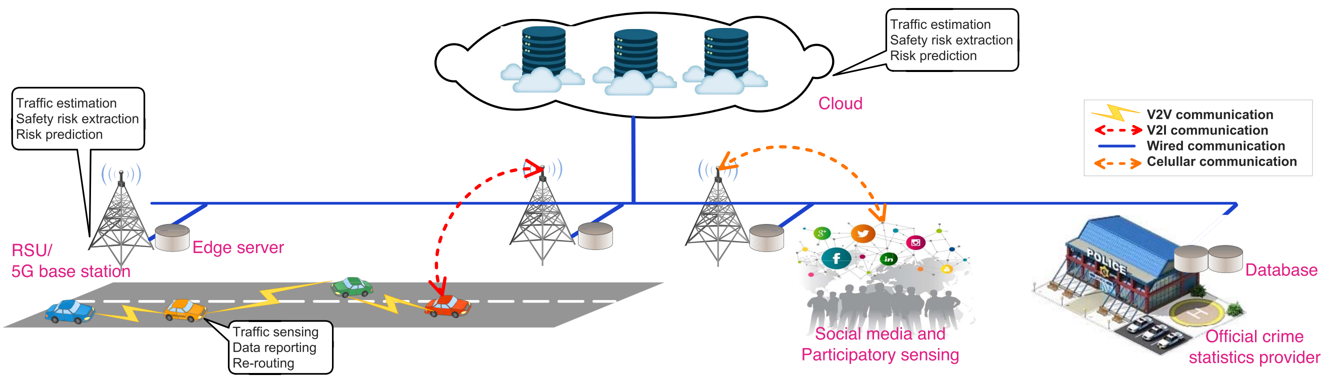


Fig. 1. The architecture employed by SNS, presenting its main components and activities.

(e.g., traffic conditions and risky areas), and perform re-routing to improve their mobility and security. On the other hand, edge servers widely deployed on roadside infrastructures can provide processing and storage resources on the network edge, leading to fast responsiveness. In this sense, each edge server is responsible not only for building local pieces of knowledge about traffic conditions in its coverage based on vehicle reports, but also for predicting future risky areas based on public safety information. Naturally, the cloud extracts both global knowledge about traffic conditions and global awareness about future risk areas to provide to vehicles an understanding beyond the edge servers' knowledge and to deal with resource constraints presented by them. Finally, the criminal-related data providers are responsible for feeding the system with criminal reports to build pieces of knowledge about city-wide illegal activities. Those providers can be: (*i*) police departments providing the history of official crime statistics; and (*ii*) people providing a more dynamic knowledge and real-time sensing based on crowdsourcing approaches (e.g., participatory sensing applications and social media), which are not possible by using just historical data.

In this scenario, vehicles can periodically re-route themselves to improve the overall traffic efficiency while decreasing safety risks according to their preferences. To do so, vehicles need to sense the urban environment and provide traffic-related information to the servers using an efficient traffic-aware data reporting mechanism detailed in Section IV. On the other hand, edge servers need to explore the spatial and temporal correlation of different criminal activities and predict future safety risk dynamics of the city using a recurrent neural network described in Section V. Therefore, with both pieces of knowledge (e.g., traffic conditions and safety dynamics), vehicles can employ an efficient personalized and cooperative context-aware traffic re-routing algorithm described in Section VI to improve their mobility while avoiding their chosen safety risks. To explain each procedure employed by the system, we defined the following scenario modeling:

Urban scenario modeling: *Considering the road network represented by a direct graph* $G = (V, E)$*, in which the set of vertices* V *represents the scenario intersections, while the set* $E \subseteq V \times V$ *corresponds to the road segments, i.e., the road segment* $uv \in E$ *represents the road segment connecting the intersections* u *and* v*. Each road segment* $uv \in E$ *has a length represented by* l_{uv} *and a shape* S_{uv}*, which is a set of points* $p \in S_{uv}$ *defining its real geographic shape. Moreover, each road segment also has two attributes* τ_{uv} *and* r_{uv}*, defining its current traffic condition and its safety risk, respectively. Each vehicle in the network is represented by the set* N*, in which each vehicle* $n \in N$ *has a pair of origin* $s \in V$ *and destination* $t \in V$*, such that* $s \neq t$*, and is associated to a path* $P \subseteq E$

connecting s to t, which defines the vehicles' route. Eventually, the traffic condition and the safety risk of a path P is defined as $\tau_P = \sum_{uv \in P} \tau_{uv}$ and $r_P = \sum_{uv \in P} r_{uv}$.

IV. Traffic-Aware Data Reporting

Vehicles need to sense the environment and report traffic-related information frequently to enable accurate traffic condition estimation and also detect congestion spots timely. Therefore, if all vehicles try to communicate their pieces of information at the same time, they can overload the network with redundant transmissions, and, consequently, degrade the overall system performance. Therefore, SNS was designed to reduce the amount of traffic information reports while producing accurate knowledge about traffic conditions.

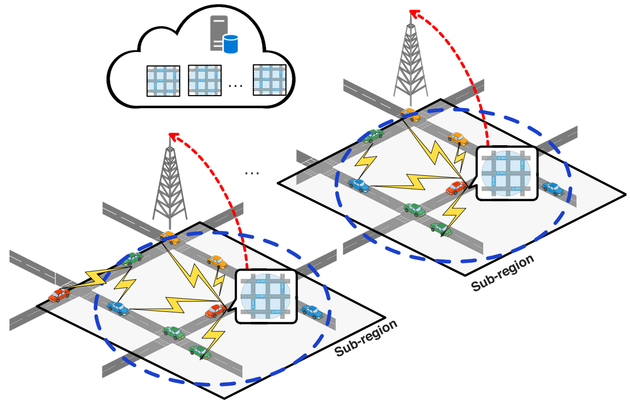


Fig. 2. Delay-based approach considering traffic density and the sub-regions.

A. Traffic Reporting

Vehicular traffic-aware reporting is based on two assumptions: (*i*) vehicles need to inform which roads are not with free-flow traffic condition, and (*ii*) vehicles in denser areas are more likely to report redundant information. In this way, we propose a density-based mechanism such that vehicles report their traffic information only when the density is higher than a predefined threshold. This approach reduces the overhead produced by the system, and also does not degrade the re-routing effectiveness, since it still provides accurate traffic estimations.

The main idea of this approach is to minimize the number of traffic reports by segmenting the road network into smaller regions named as sub-regions. Then, we select only the best vehicles within each sub-region to report their traffic estimations of the whole sub-region. The road network segmentation is a trade-off between the accuracy of knowledge about traffic conditions and the number of traffic reports. Thus, the higher the size of each sub-region, the lower the amount of traffic information reported to the main server. However, each sub-region must be defined carefully because the larger its size, the harder to estimate its traffic conditions accurately. Exploring the best size and shape of each sub-region is beyond the scope of this work. Thus, we used the segmentation presented in [13], which takes into account the vehicle's communication range to define each sub-region equally.

SNS sends traffic reports according to the following requirements: (*i*) traffic reports are sent periodically or whenever a vehicle notices that the density of a particular road in its sub-region can potentially evolve to congestion; and (*ii*) SNS chooses which vehicle will be responsible for reporting its traffic estimation by using a delay-based approach as a function of the vehicles' density in its sub-region.

The vehicles' density is computed locally by each vehicle in the sub-region based on the pieces of information shared by its neighbors (within the same sub-region) using periodic beacon exchange. Each vehicle emits beacons with the same frequency, in which each beacon has the vehicle identifier, vehicles' current road segment, and velocity. Thus, by counting the number of received beacons in a short time window (e.g., 5 seconds), each vehicle can estimate the density of its vicinity.

The delay-based approach enables that each vehicle in the same sub-region schedules a traffic report based on a computed delay. To avoid redundant transmissions, if a vehicle overhears another report from the same sub-region during its waiting time (i.e., according to its delay), it cancels its report. The delay is computed based on the degree of each vehicle (e.g., number of neighbors), defined as:

$$d_n = -\left(\frac{\max(d)}{\max(\delta)}\right) \cdot \delta(n) + \max(d) \tag{1}$$

where, $\delta(n)$ is the neighbor degree of vehicle $n \in N$, while $\max(\delta)$ is the maximum neighbor degree of its sub-region, which is estimated based on the received beacons, and $\max(d)$ is a parameterized value to define the maximum delay to report a traffic estimation to the main server.

This approach considers the number of neighbors to prioritize vehicles with better estimations. Since in our implementation the sub-regions have about the same size as the communication range of the vehicles, the vehicles with higher degree potentially will have a better traffic estimation than the others. For the sake of clarity, Figure 2 shows the previously described delay-based approach to reduce the number of traffic report transmissions. In particular, after estimating the traffic condition of the roads under its coverage, each vehicle schedules a traffic report using the delay-based approach (considering the number of vehicles in its vicinity). In Figure 2, the vehicle close to the center of each sub-region has a higher degree (i.e., vehicles in its vicinity) than the others. Thus, they are responsible for sending their traffic estimations to the server. Finally, when the other vehicles hear this communication they will cancel their transmission to avoid redundant transmissions.

B. Traffic Condition Estimation

The server receives the reports from vehicles containing the number of vehicles on each road segment for every road on their sub-regions. Whenever the server receives a report concerning a sub-region, it will compute an exponential moving average [30] for each road in that sub-region. In this sense, the server estimates the traffic condition considering two cases:

Free-flow estimation: for the road segments without any traffic reports, the estimation is considered as free-flow, which is equal to the travel time spent to travel the entire road segment.

Traffic condition estimation: for the roads with density greater than 0, the traffic condition estimation is based on the Greenshield model [4], which considers the relation between speed and density. This model is extensively used by transportation researchers [30], [39] and was shown empirically to describe well the speed-density relation for relatively low densities. It considers a linear relationship between the estimated average speed and the traffic density on each road segment as follows:

$$\operatorname{avg} s_{uv} = \max s_{uv} \left(1 - \frac{N_{uv}}{\frac{l_{uv}}{\operatorname{len}(n)+\beta} \cdot lanes} \right) \tag{2}$$

$$\tau_{uv} = \frac{l_{uv}}{\operatorname{avg} s_{uv}}, \tag{3}$$

where, N_{uv} is the current vehicles' density, $\frac{l_{uv}}{\operatorname{len}(n)+\beta}$ is the maximum density of the road uv computed based on the length of the road l_{uv}, the average vehicle's length $\operatorname{len}(n)$, the minimum gap β between each vehicle in each lane of the road uv and the number of lanes of such road. The average speed $\operatorname{avg} s_{uv}$ is estimated according to the model and considering its maximum speed $\max s_{uv}$.

V. Discovering Risky Areas and Exploring Spatiotemporal Information

Criminal activities are motivated by different environment-related dynamics [3], [20]. Therefore, the same region can provide different conditions throughout the day to either increase or decrease criminal opportunities within it. In other words, criminal activities have a spatiotemporal correlation, which produces hotspots (i.e., regions with high number of crimes) for specific criminal activities along the day. People are the main targets of criminal activities and they have daily routines [20]. This relation produces spatiotemporal patterns at the locations of risky areas. A risky area in this work is considered as a region that is likely to have an elevated number of criminal incidents according to the time and the day.

In this way, based on the displacement of risky areas along the day, we can get relevant information to aid traffic management systems for providing safer routes. This section describes the methodology used to obtain spatial and temporal correlations and also forecast risky area locations based on day and time according to each criminal incident (e.g., robbery, assault, narcotics, weapons violation, and kidnapping).

A. Exploring Spatial and Temporal Correlation

To identify the spatiotemporal correlation, the spatial density of each criminal activity was computed based on the geographic coordinates of each criminal incident considering a predefined time window (t_{start} and t_{end}). In this way, let C be a set of criminal incidents that happened in the city obtained either from an official database of criminal reports or even through social sensing (e.g., social networks and participatory sensing applications). Each criminal incident $c \in C$ has its location (e.g., geographic coordinates), date, time and type (e.g., assault, robbery, narcotics, weapons violation, and kidnapping). Thus, the spatial density for each criminal activity was defined based on a Gaussian Kernel Density Estimation (KDE), in which based on a set of C criminal incidents related to a particular type of criminal incident happened between a time window t_{start} and t_{end} the KDE estimates the criminal density at a point p as:

$$\theta(p) = \frac{1}{|C|} \cdot \sum_{c \in C} \frac{1}{h\sqrt{2\pi}} e^{\left(-\frac{1}{2}\left(\frac{||c-p||}{h}\right)^2\right)} \tag{4}$$

where $||c - p||$ is the Euclidean distance between the points c and p and h is the bandwidth used. The bandwidth h defines the Gaussian kernel spread, which controls the smoothness of the estimated density, according to [34]. Once the density function is estimated, the criminal activity density of each road for each crime can be computed based on its shape, defined by:

$$\sigma(uv) = \sum_{p \in S_{uv}} \theta(p), \tag{5}$$

where, S_{uv} is the set of points p that define the shape of the road segment $uv \in E$. Thus, the criminal activity of each road segment can be obtained as the normalized densities:

$$r_{uv} = \frac{\sigma(uv)}{\sum_{u'v' \in E} \sigma(u'v')}, \tag{6}$$

where r_{uv} is proportional to the probability of observing a crime incident at road $uv \in E$. Analogously, the criminal activity of a path P is r_P which describes the total criminal density of the path, e.g., $r_P = \sum_{uv \in P} r_{uv}$.

In this context, to explore the spatial and temporal correlation of each criminal activity along the day, we defined the time window size to be 1 hour and then compute the KDE for all roads considering the set of crimes happened in that period for each type of criminal activity. Therefore, we can identify the hotspots for each type of crime (e.g., assault, robbery, and narcotics) in each hour of the day.

B. Predicting Criminal Densities

The key idea in predicting criminal densities is to learn the safety risk dynamics in each neighborhood to know in advance when some region will be a dangerous place considering a specific type of crime and its spatiotemporal correlation. To learn such dynamics, the system needs to be able to extract the knowledge from previous events [33]. In this context, several machine learning techniques have been used to pull out the knowledge obtained from past events such as linear regression, Markov models, support vector machines, and neural networks [41]. However, these approaches potentially have issues considering spatial and temporal setup, since they can not distinguish the spatial and temporal correlations accurately [25], [33].

Recurrent neural networks (RNN) is a deep learning approach that extends the traditional feed-forward networks

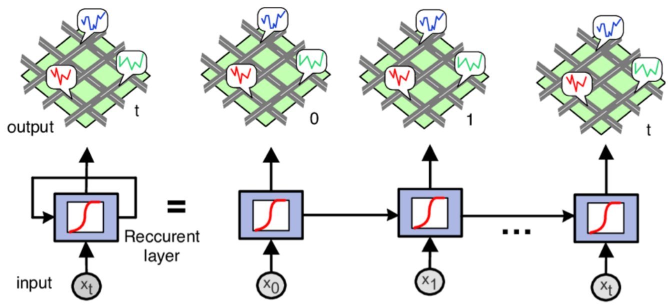


Fig. 3. RNN employed by SNS.

with internal cycles [40]. These internal cycles support recurrent neural networks with an internal state that is capable of tracking sequences of information and learning relevant spatial and temporal features by exploring intercorrelation among current and past inputs. The long short-term memory (LSTM) neural network architecture is widely used due to its resistance to the vanishing gradient problem and ability to handle long-term dependencies [28].

To employ an RNN in SNS, we first need to build a dataset extracted from the criminal densities such as the estimations based on the crimes happened in the city of Chicago, during 2018 (available at Chicago Open Data Portal[2]) and then provide the data input to the RNN to be able to perform the predictions. In this way, let $X = \{x_0, x_1, \ldots x_m\}$ be a vector that represents the dataset, in which each element $x_i \in X \mid 0 \leq i \leq m$ consists of a tuple $x_i = \langle timestamp, uv, r_{uv}, crime\rangle$, representing a date and time for the estimated criminal density of the road segment uv for a specific crime. The predictor defined as $f = \Psi \circ \Phi$, where $\circ$ indicates applying function Ψ on function Φ's output. The input data first goes through the feature learning machine $\Phi(\cdot)$, which is used to transform inputs into features. The second step involves the representation function $\Psi(\cdot)$, which maps features into a prediction [25]. Thus, the prediction process of the next criminal density for each road segment and type of crime is given by:

$$x_{uv}^{t+1} = \Psi_{uv} \circ \Phi_{uv}(X_{uv}^{t}) \tag{7}$$

where, x_{uv}^{t+1} represents the next criminal density of the road segment uv considering its previous observations X_{uv}^{t}.

Figure 3 shows how the RNN employed by SNS works. The RNN receives an input x^t representing a set of tuples (e.g., timestamp, uv, r_{uv}, type of crime) and for each road segment and type of crime it employs a LSTM as recurrent layer with a tanh activation function. We illustrate how the recurrent layer works by unrolling the past predictions $x_0, x_1, \ldots, x_t$, which are used as long-term memory to explore the internal correlation between each prediction. At last, the output is represented as the future safety risk dynamic of each road considering each type of crime. It is worth noticing that we considered the future safety risk dynamic as the criminal density for the next hour (e.g., we used granularity of 1 as time window to compute the criminal density of each road using Equation 6). In other words, the RNN employed by SNS provides the criminal density for each road for the next hour considering each type of crime.

[2]https://data.cityofchicago.org

VI. Cooperative Safety-Aware Re-Routing Algorithm Based on Drivers Preference

To enable personalized safety-based re-routing, each vehicle needs to be able to compute its route according to its preference (e.g., based on which crime each driver wants to avoid). Therefore, SNS offloads the route computation from the cloud and edge server to each vehicle, which also improves system scalability [30].

The effectiveness of a vehicular traffic re-routing algorithm is directly related to its ability to balance the traffic flow to avoid the creation of different congestion spots. Thus, when computing an alternative route for some vehicle, the re-routing algorithm must be aware of the routes previously taken by the other vehicles in the same time window (i.e., re-routing interval). However, by offloading the route computation in each vehicle (e.g., distributed approach) this task becomes not as straightforward as in centralized ones. In distributed approaches, each vehicle needs to inform the others about its route whenever a new route is computed. Yet, spreading all routes of all vehicles in a multi-hop approach to the whole network during every re-routing phase is not feasible, since it will produce high network load as the density of vehicles increases [30]. Besides, since each vehicle needs to wait for the routes from the other vehicles to compute its route, it might introduce high latency to the system if not adequately addressed.

In this scenario, when the server detects signs of congestion on any road, it will alert the vehicles by sending the updated traffic view (e.g., current traffic conditions on the roads previously estimated) and the future safety risks predicted (e.g., the next criminal density of each road for each type of crime) containing all roads that have different values (traffic condition and safety risk) from the last update. The server sends the pieces of knowledge only to the vehicles that reported the sub-region traffic estimation most recently. This process triggers the re-routing, which is composed of two main phases: (*i*) knowledge dissemination; and (*ii*) cooperative route computation. The first phase provides the updated pieces of knowledge about traffic conditions and safety risks to the vehicles that potentially will be affected by the congestion, while the other one is responsible for optimizing the re-routing algorithm to achieve better traffic management. The cooperative re-routing algorithm needs to consider the following issues: (*i*) how to properly inform vehicles about the routes of the others, and (*ii*) how to balance the traffic based on the information of its neighborhood considering their different preferences related to safety risks.

A. Cooperative Route Sharing

The re-routing algorithm proposed in SNS focuses on balancing the traffic flow over a set of alternative routes for each vehicle based on their current position, destination, and its preference (e.g., type of crime that each driver wants to avoid). The set of alternative routes are computed based on the Pareto

Algorithm 1 Cooperative Route Sharing Algorithm

```
Input : G // Road network graph
   1 T // Traffic condition knowledge estimated by SNS
   2 C // Safety risk knowledge predicted by SNS for
     each type of crime
Output: Broadcast of the set of alternatives paths of the vehicle and its priority
   // Update the knowledge about traffic conditions and
      safety risk on the roads and disseminate it through
      the network using a data dissemination protocol
3  G.updateTrafficKnowledge(T);
4  disseminate(G);
   // Get the current route of the vehicle
5  P ← getRoute();
   // Get the origin, destination and the preference of the
      vehicle
6  s ← P.getCurrentRoad();
7  t ← P.getDestination();
8  C ← selectCrimeToAvoid(C);
9  G.updateSafetyRiskKnowledge(C);
   // Compute the set of alternative routes from s to t that
      optimizes both mobility and safety risk according to
      the preference of the vehicle
10 𝒫 ← G.computeParetoSet(s, t, C);
   // Compute priority based on the distance to the nearest
      congested road
11 rank ← computeRank(P, G);
   // wait rank ms to broadcast its alternatives paths, its
      priority and its preference related to the type of
      crime
12 wait(rank);
13 broadcast(𝒫, rank, C);
```

set (which is the set of paths that optimizes both mobility and safety). However, to effectively provide this approach in a distributed way, the major concern to be addressed is the latency. In this sense, we propose a cooperative route sharing approach considering that:

- Vehicles need to know as many routes as possible taken by their neighbors to achieve a better route guidance.
- Vehicles cannot wait to compute their routes after all their neighbors.

To tackle these concerns, we propose a rank-based approach that gives higher priority to vehicles closer to the congestion to compute their routes earlier than the others, assuming that those vehicles are the ones who need a faster response. The rank is a delay calculated as the function of the distance between the vehicle and the next congested road in its route. This rank-based approach minimizes the broadcast storm problem during route sharing since it avoids that vehicles share their routes at the same time. Also, it reduces the latency, because vehicles only wait for the routes from vehicles with higher ranks rather than all their neighbors.

Algorithm 1 describes the cooperative routing sharing approach. In summary, when a vehicle receives some new information about traffic conditions and safety risks, it updates its traffic knowledge (Line 3) and forwards it using an efficient data dissemination protocol, which employs a delay-based broadcast suppression mechanism to reduce the overhead while maintaining a high delivery ratio and a low delay [14]. Then, it computes the set of alternative routes (e.g., the Pareto set $\mathcal{P}$, described in Section VI-B) based on its current position, destination, and its preference regarding the type of crime to avoid. Also, it computes its *rank* based on the distance between its current position and the nearest congested road (Lines 6-11). Finally, according to its rank, the vehicle waits to broadcast its set of alternative routes, its rank and its preference related to the type of the crime that it wants to avoid to its neighbors (Line 13).

Each vehicle shares its set of alternative routes and its rank to enable that other vehicles infer which route each vehicle has selected based on all routes received and the rank of each neighbor.

B. Computing the Pareto Set

The Pareto set $\mathcal{P}$ for a vehicle with origin s and destination t, is a set of paths that optimize both metrics mobility and safety risk (e.g., τ_P and r_P). Formally, let P_{old} be the current route of a vehicle (assumed to be the shortest path) and P_{new} be a path in the road network G linking s to t. Thus, the Pareto set will be composed by the set of paths that respect the following condition:

$$\mathcal{P} = \begin{cases} P & \text{if } \tau_{P_{\text{new}}} \leq \tau_{P_{\text{old}}} \text{ and } r_{P_{\text{new}}} \leq r_{P_{\text{old}}}, \quad \forall P_{\text{new}} \in G \\ \emptyset & \text{Otherwise} \end{cases} \tag{8}$$

The safety risk of a path P is defined as $r_P = \sum_{uv \in P} r_{uv}$. Thus, considering $r_P = \lambda$ the safety risk of the new path must be a number between $[0, \lambda]$. In this scenario, the problem of finding the Pareto set of a path with origin s and destination t, is similar as to find the fastest path P with $r_P \leq \varphi$ such that $\varphi \in \{0.01, 0.02, \ldots, \lambda\}$, thus, satisfying the following recurrence:

$$T(v, \varphi) = \begin{cases} 0 & \text{if } \varphi = 0 \text{ and } v = s, \\ \infty & \text{if } \varphi = 0 \text{ and } v \neq s, \\ \min \begin{cases} T(v, \varphi - 0.01) \\ \min\limits_{u : r_{uv} \leq \varphi} \{T(u, r_{uv} - 0.01) + \tau_{uv}\} \end{cases} & \\ & \text{Otherwise,} \end{cases} \tag{9}$$

Such recurrence naturally derives a recursive algorithm with exponential time complexity, but it also can be solved using a dynamic programming approach having a complexity of $O(|E|\lambda)$, which has a pseudo-polynomial complexity since in our scenario, the value of λ is not arbitrarily large, since the safety risk of each road is at most 1.

Algorithm 2 describes the dynamic programming approach used for computing the Pareto set. In the algorithm, DP is a dynamic programming table used to memorize the traffic efficiency value to reach each vertex v from a vertex s with risk φ, for $s, v \in V$. Such a path can be stored by table ψ, in which $\psi[v, \varphi]$ is a predecessor of v in the (s, v)-path with risk φ. The first and the second loops (lines 3-10) are responsible for preparing the DP and ψ tables in the base cases of the recurrence, which mark empty paths by clearing the predecessor of source s, and mark (s, v)-paths as unfeasible whenever the demanded risk is $\varphi = 0$. Later, the algorithm fills up the table DP, for the following risk values $\varphi \in \{0.01, 0.02, \ldots, \lambda\}$: the best traffic efficiency is either the same as that of risk $\varphi - 0.01$ (line 14), or is achieved by following some non empty (s, v)-path from s to some predecessor u, and edge uv (lines 15-22). Possible predecessors are those vertices u for which $r_{uv} \leq \varphi$, and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 2 Dynamic Programming Algorithm for Computing the Pareto Set

```
Input  : s // current position of vehicle n
         t // destination of vehicle n
         λ // maximum risk of the path
Output: Pareto set of paths stating in s and ending in t with risk at most λ.
1  PD ← [ ] ; // table to store T(v, φ)
2  ψ ← [ ] ; // stores predecessor vertex in the route PD[v, φ]
3  foreach φ ∈ {0.01, 0.02, ... λ} do
4      ψ[s, φ] ← ∅;
5      PD[s, φ] ← 0;
6  end
7  foreach v ∈ V \ {s} do
8      ψ[v, 0] ← ∅;
9      PD[v, 0] ← ∞;
10 end
11 foreach φ ∈ {1, 2, ..., λ} do
12     foreach v ∈ V do
13         ψ[v, φ] ← ψ[v, φ − 0.01];
14         PD[v, φ] ← PD[v, φ − 0.01];
15         foreach uv ∈ E do
16             if r_uv ≤ φ then
17                 if PD[v, φ] > PD[u, r_uv − 0.01] + τ_uv then
18                     ψ[v, φ] ← u;
19                     PD[v, φ] ← PD[u, r_uv − 0.01] + τ_uv;
20                 end
21             end
22         end
23     end
24 end
25 return PD[t, ], ψ;
```

Algorithm 3 Updating Alternative Route

```
Input  : G // Pieces of knowledge about traffic conditions
           and safety risk
       1 Q // The set of messages containing the
           alternative routes, the rank and its preference of
           each neighbor
Output:
   // Sort Q according to the rank of each vehicle
2  Q.sort();
   // Get its own rank
3  r ← getMyRank();
4  while Q is not empty do
5      q ← Q.pop();
       // Check if its neighbor has a greater rank
6      if q.rank < r then
           // Get the set alternative paths of its neighbor
7          𝒫 ← q.getRoutes();
           // Infer the route taken based on the current
              traffic and safety knowledge and the preference
              of the vehicle
8          P ← selectRoute(𝒫, G, C);
           // Update the traffic knowledge
9          G.update(P);
10     end
11 end
   // Compute the least popular route based on the pieces
      of information shared by its neighbors
12 𝒫 ← myAlternativePaths();
13 P ← selectRoute(𝒫, G);
14 setRoute(P);
```

the corresponding (s, v)-path in this case has traffic efficiency $DP[u, r_{uv} - 0.01] + \tau_{uv}$ (line 19). Finally, the Pareto set can be obtained through line t for all values of φ.

C. Cooperative Context-Aware Re-Routing

In our previous work, we proposed EBPOP [10], which aims to distribute the traffic flow over the routes available from the Pareto set. In general, the key idea is to perform an intelligent path selection considering the risk of each road segment and the number of vehicles going toward such road. Unlike deterministic approaches for multi-objective optimization such as weighted-sum [27], resource constrained shortest path [21], and evolutionary algorithms [26], EBPOP is able to provide better traffic management since it provides an efficient traffic balance even considering multiple metrics [10]. Since EBPOP was designed to work with a centralized architecture, we present some changes introduced to enable a cooperative re-routing (e.g., enabling EBPOP to work as a distributed algorithm in SNS as shown in Algorithm 3) and enable personalized safety risk re-routing.

To avoid creating new congestion spots, a popularity measure is associated to each road segment. A new congestion spot might come up if many vehicles take the same road segment within the same future time window. Hence, we define the demand d_{uv} of a road segment $uv \in E$, as the number of vehicles that have the road uv included into their path.

By using the entropy of information theory, each vehicle can compute the entropy of each path P of the Pareto set $\mathcal{P}$ received from its neighborhood, given by the function $\xi(P)$:

$$\xi(P) = -\sum\nolimits_{uv\in P} \frac{\text{avg}(r)^t \cdot d_{uv}}{r_{uv}\sum_{uv\in E} d_{uv}} \log \frac{d_{uv}}{\sum_{uv\in E} d_{uv}}, \quad (10)$$

where, $\text{avg}(r)^t$ is the average road risk estimation in time t, r_{uv} and d_{uv} is the risk and the demand of the road segment $uv \in E$, respectively.

Moreover, based on the entropy of each path each vehicle estimates the popularity of the path based on $e^{\xi(P)}$. Thus, focusing on providing a better distribution of the traffic flow, the route selected for each vehicle is given by the least popular path of the Pareto set. In other words, P is defined as:

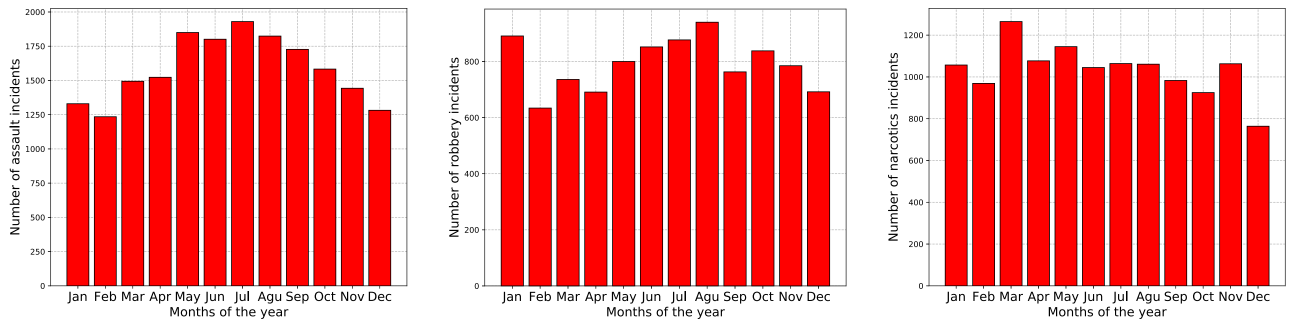
$$P = \arg\min_{P\in\mathcal{P}} \left(e^{\xi(P)}\right), \quad (11)$$

where, P is an $(s\text{-}t)$-path and $\mathcal{P}$ is the set of paths that compose the Pareto set.

Upon selecting a path for a vehicle, the demand of all roads uv that compose the path P are updated, in order to estimate the popularity of each path. For the sake of clarity, Algorithm 3 describes the whole procedure that a vehicle executes to select its route based on its neighborhood information. To compute its route, each vehicle first sorts the set of routes received from its neighborhood according to each rank (Line 2). Then, it compares its rank with the others and to each vehicle with a higher priority, it infers the route selected by the vehicle using the current demand information, and its preference (e.g., type of crime that each vehicle wants to avoid) using the Equation 11 (Lines 7-8). Thereafter, it updates the demand d_{uv} of each road $uv \in P$ in the route selected by the vehicle (Line 9). Finally, after inferring the routes taken by its neighbors with higher ranks, the vehicle computes its own route avoiding popular routes (Lines 12-14).

VII. Performance Analysis

This section analyzes the performance of SNS concerning its architecture, network performance, personalized traffic re-routing with spatiotemporal correlation, and criminal



(a) Assault (b) Robbery (c) Narcotics

Fig. 4. Criminal incident distribution of each month during 2018 considering Assault, Robbery and Narcotics incidents.

density prediction. Subsection VII-A introduces the simulation platform, describing the tools, scenario and analyzed metrics. Subsection VII-B describes the results of the criminal density predictor for each type of crime used by SNS. Subsection VII-C compares SNS with our previous work EBPOP in respect to its network mobility and safety performance. At last, Subsection VII-E shows the results of the personalized re-routing algorithm.

A. Methodology

The simulation platform is composed by the simulator of urban mobility, SUMO [5], version 0.30.0, the network simulator OMNeT++ [38], version 5.0 and also the vehicular networking framework Veins [36], version 4.6. The road network is composed by a fragment of 50 km^2 from Chicago, obtained using OpenStreetMap. The traffic mobility was produced using the TrafficModeler [31] tool to ensure a realistic traffic mobility, resulting in a total of five thousand routes. The number of routes was defined to create heavy traffic and congestion (i.e., the travel time for a majority of the cars is significantly higher than the free flow travel time). In this way, we generate the traffic at a constant rate by deploying one car each second in the simulator from one side of the scenario to the other one. By default, the shortest travel time paths are automatically calculated and assigned to each vehicle at the beginning of the simulation based on the road speed limits. Moreover, to implement the RNN we used the TensorFlow framework version 1.12 [1]. The presented results have a confidence interval of 95% and Table II shows additional parameters used in the simulation.

TABLE II

SIMULATION PARAMETERS

Parameters	Values
Channel frequency	5.890e9 Hz
Propagation model	*Two ray*
Transmission power	2.2 mW
Communication range	300 m
Bit rate	18 Mbit/s
PHY model	IEEE 801.11p
MAC model	EDCA
Max hop count	10
Scenario	Chicago
Scenario size	50 km^2
Re-routing interval	450 s
Recurrent Layer	Bidirectional LSTM
Activation function	tanh
Loss function	Mean square logarithmic error
Batch size	128
Sequence length	32
# of LSTM cells	128
EPOCS	100

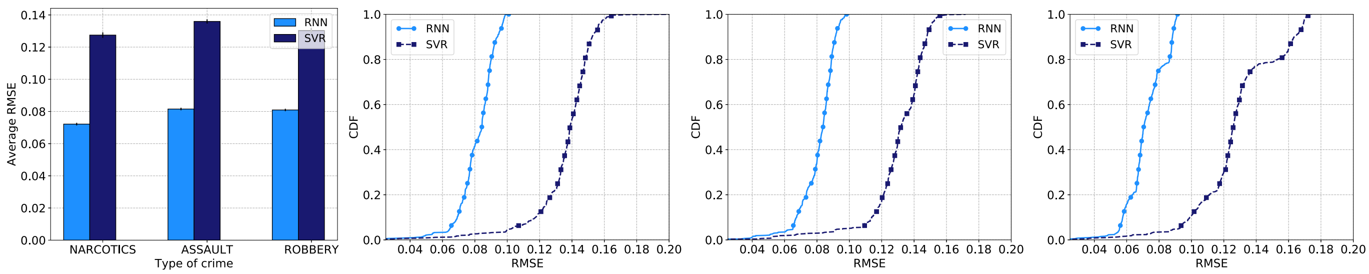
B. Criminal Density Prediction

The criminal density prediction analysis verifies how close the predictions provided by the RNN implemented in SNS are in respect to the real density extracted from the dataset for each type of crime. To analyze the RNN performance, we used 70% of the samples of the data for training and 30% for testing considering the three types of crimes with higher frequency in the dataset classified as Assault, Robbery and Narcotics. For more information please refer to Chicago Open Data Portal. Figure 4 shows the distribution of each type of crime used in our evaluation for each month during 2018.

In this evaluation, each prediction represents the criminal density of each road for each type of crime with granularity of one hour considering the previous events. In other words, to predict the criminal density of the next hour the RNN exploits the behaviour of the criminal density of each road in the past to understand its intercorrelation through LSTM cell and improve the prediction [25], [40]. At last, performance of the RNN was measured in terms of root mean square error (RMSE) in comparison to supervised vector regression (SVR).

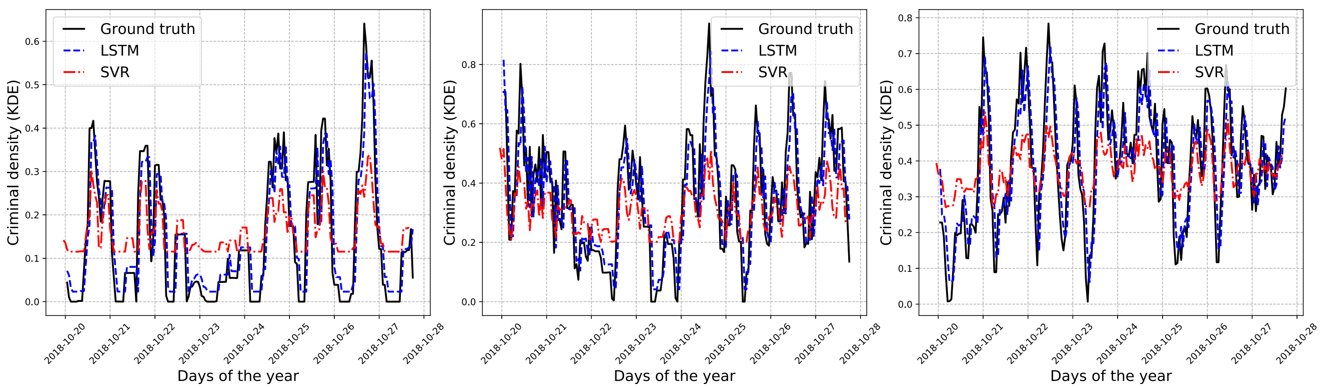
Figure 5 shows the RMSE results for each type of crime considering RNN and SVR. In particular, Figure 5(a) shows the average results for RMSE, while Figures 5(b), 5(c), and 5(d) show the results for RMSE as a Cumulative Distribution Function (CDF) for assault, robbery and narcotic crimes, respectively. The predictions provided by the RNN achieves an average RMSE of approximately 0.08 for all crimes considering all roads segments, while the predictions of SVR reaches an average RMSE of about 0.14. This is the result of the efficiency of RNN in learning the internal correlation of past events and how it can impact future predictions and also its ability to exploit the spatiotemporal correlation, which improves its performance in at least 40% when compared to SVR (see Figure 5(a)). Moreover, we can see that the highest error in the predictions is approximately 0.1 (see Figure 5(b), considering the same unit of safety risk estimation), which shows that even in case of mispredictions the risk potentially introduced in the new route is lower

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Average RMSE (b) Assault (c) Robbery (d) Narcotics

Fig. 5. Results of criminal density predictions provided by the RNN and SVR.



(a) Narcotics (b) Robbery (c) Assault

Fig. 6. Comparing criminal density predictions provided by the RNN with real criminal density extracted from the dataset during one week of October 2018.

than 0.1. On the other hand, the highest error of SVR is at least 70% higher than the error of RNN, which might potentially decrease the system performance due to false positives.

In case of overestimation (e.g., the real safety risk on the road is lower than the predicted one) this is not an issue to SNS, because it potentially results in safer routes. On the other hand, in case of underestimation (e.g., the real safety risk on the road is higher than the predicted one) the risk introduced is not harmful since the road segment can be at most 0.1 riskier than predicted when using RNN predictions. In other words, SNS will not guide vehicles towards high risk roads assuming that those roads are safe, which is important for the reliability of the system.

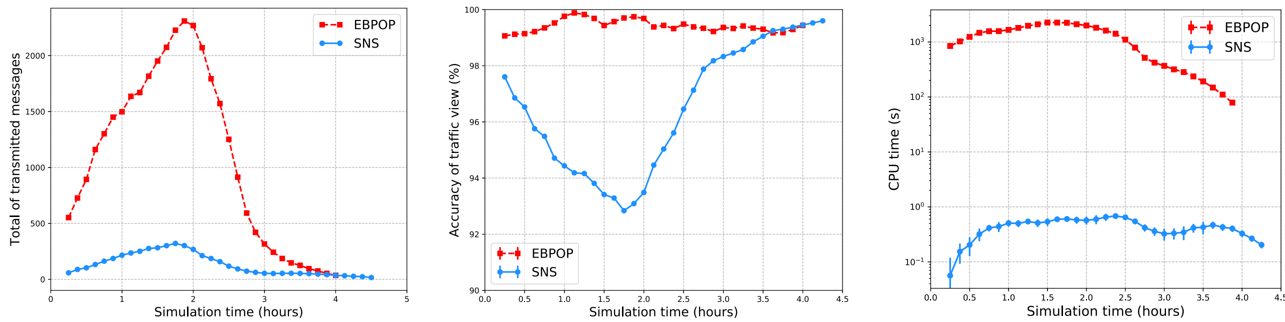
For the sake of understanding, Figure 6 compares the criminal density prediction of RNN and SVR for all crimes with its real criminal density extracted from the dataset during one week of October 2018 in the center of Chicago. As it can be seen, the RNN is able to express the same behavior of the real criminal density all day long for all crimes. Moreover, we can observe that the predictions are close to the real values and underestimations are not frequent. On the other hand, SVR does not express a behavior as accurate as RNN, which could potentially degrade the SNS performance due to frequent underestimations. Therefore, based on these results we can conclude that: (*i*) the RNN provides accurate predictions and can understand the spatiotemporal information; (*ii*) the error introduced in case of mispredictions is low; and (*iii*) the RNN is a suitable and reliable approach for predicting safety risks.

C. Network Overhead, Traffic Management and Safety Risk Analysis

With this analysis we evaluate the SNS performance in terms of network cost, system scalability, complexity time, traffic efficiency and safety risk in comparison with EBPOP. In this way, the following metrics were assessed:

- **Transmitted messages**: the total number of transmitted messages in the system to deliver its service. This metric includes traffic reporting transmissions as well as route recommendation and traffic view dissemination. A high number of messages transmitted is a strong indication of redundant and unnecessary transmissions.
- **Traffic estimation accuracy**: the percentage of roads with estimated traffic conditions equal to the real one (extracted from the simulator) at each re-routing interval. Low accuracy means that either the proposed traffic reporting is not a good mechanism for representing the traffic conditions on the roads or the multi-hop data dissemination protocol is not able to deliver the traffic view to the target vehicles (e.g., vehicles that need to receive the traffic view to improve their mobility).
- **CPU time**: is the time spent by the re-routing algorithm to compute the new alternative route to the intended vehicles. A longer CPU time potentially introduces latency to the system, consequently degrading its overall performance.
- **Relative travel time**: the ratio between the vehicle's travel time using one of the re-routing approaches and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Transmitted messages (b) Accuracy of the traffic knowledge (c) Complexity

Fig. 7. Network and time complexity results comparing EBPOP and SNS.

its travel time without any re-routing solution at all. This metric summarizes the travel time reduction for each vehicle.

- **Relative safety risk**: the ratio between the safety risk of the vehicle's route using one of the re-routing approaches and its route safety risk without any re-routing solution at all. This metric summarizes the safety risk minimization for each vehicle.

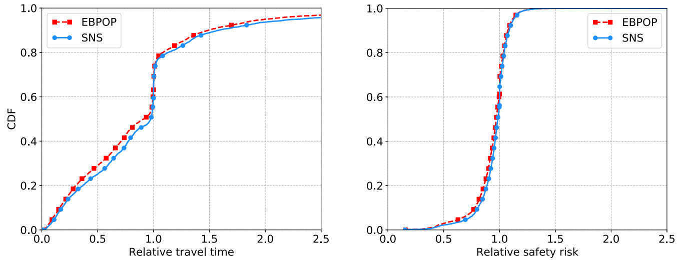
Figure 7 shows the results for the network cost, system scalability and time complexity metrics for SNS and EBPOP. With this evaluation, we want to assess how the cooperative re-routing algorithm and the hybrid architecture behave concerning the centralized one. Figure 7(a) shows the transmitted messages as a function of the simulation time according to each re-routing step. As expected, the EBPOP transmits substantially more messages than SNS since it does not implement any mechanism to reduce the number of traffic reports. Thus, each vehicle needs to report its traffic measurements to the server periodically (e.g., in each re-routing step). Moreover, due to the centralized approach implemented by EBPOP, whenever a vehicle passes through a congestion (e.g., identified by the server), the edge server needs to send a message with the already computed alternative route to such vehicle using the closest RSU. Hence, such fact not only further increases the number of transmissions, but also reduces the system scalability, because this procedure potentially introduces undesired latency to the system in high density scenarios. On the other hand, due to the efficient traffic reporting mechanism employed by SNS, it is able to reduce the number of transmissions to deliver its service. In particular, compared to EBPOP, SNS reduces the number of transmission in about 85%, on average. In addition, like EBPOP, SNS increases the number of transmissions according to the density of vehicles. However, such increase is much lower when compared to the increase presented by EBPOP, which means that SNS provides a better system scalability. In particular, during the peak density in our simulations (see values with simulation time between 1 and 2 hours in Figure 7(a)), EBPOP transmits about 2500 messages to deliver its service, while SNS transmits less than 400 messages.

Having an accurate knowledge about traffic conditions is essential to understand the traffic dynamics and enable a good traffic management. Figure 7(b) shows the results for the traffic knowledge accuracy in each re-routing interval for both EBPOP and SNS. EBPOP provides an upper bound close to 100% for all re-routing steps, considering that the server knows the current position and velocity of all vehicles. However, since in SNS the server receives traffic estimations of each sub-region, which is cooperatively built by the vehicles, it does not precisely know the position and velocity of all vehicles. Consequently, this decreases its knowledge about traffic conditions on the roads. In particular, in SNS, as the density increases, the accuracy decreases. Nevertheless, it still reaches an accuracy higher than 94% for all re-routing steps, which is a suitable accuracy for providing a good traffic management. In this scenario, we can see the efficiency of the traffic reporting mechanism employed by SNS, which produces low overhead and still provides an accurate knowledge about traffic conditions. On the other hand, EBPOP produces an overhead more than 5 times higher than SNS in order to increase less than 5% of the traffic knowledge (see Figures 7(a) and 7(b)).

To evaluate the time complexity we considered the CPU time of each solution. For this evaluation, we counted the amount of time spent in each re-routing step using an Intel(R) Core(TM) i5-5257U CPU with 2.70 GHz. Figure 7(c) shows the results for this evaluation. As it was expected, EBPOP presents the highest CPU time for all re-routing steps, which is a consequence of its centralized architecture, since the server needs to compute the alternative routes for all intended vehicles. Compared to EBPOP, SNS reduces in approximately 99% the CPU time (i.e., the complexity of the re-routing algorithm), due to its hybrid architecture and its efficient cooperative re-routing algorithm. At last, it is important to notice that, for EBPOP, the CPU time is directly related to the traffic density. Therefore, it potentially can introduce high latency to the system in high density scenarios and degrade its performance. On the other hand, in SNS, its CPU time has no relation with the traffic density, because it offloads the re-routing algorithm in each vehicle, consequently providing a better system scalability.

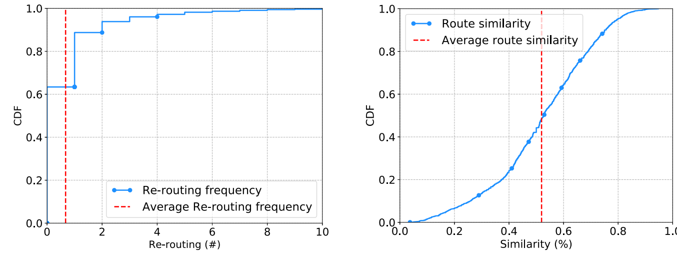
The efficiency of the cooperative re-routing algorithm employed by SNS can be seen by comparing the relative travel time and relative safety risk results. Figure 8(a) shows the relative travel time for EBPOP and SNS as a Cumulative Distribution Function (CDF). When compared to the No Re-routing approach, both solutions improve the mobility

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



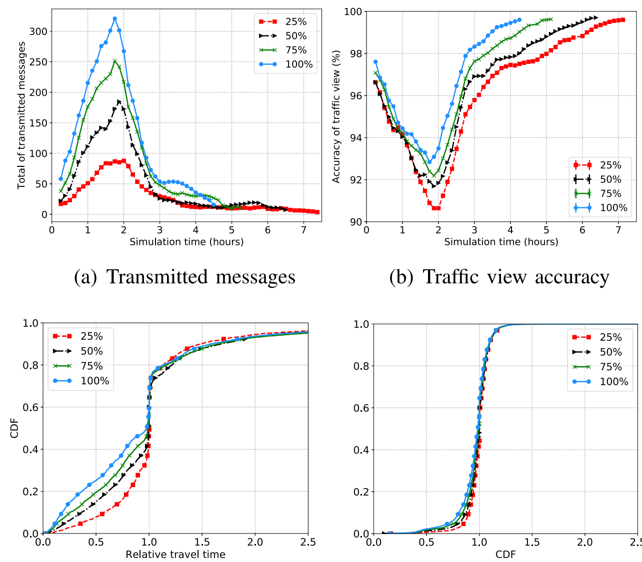
(a) Relative travel time (b) Relative safety

Fig. 8. Traffic and safety risk results comparing EBPOP and SNS.



(a) Re-routing frequency (b) Route similarity

Fig. 9. Re-routing efficiency of SNS.



(a) Transmitted messages (b) Traffic view accuracy

(c) Relative travel time (d) Relative safety

Fig. 10. SNS effectiveness based on penetration rate.

and the safety for the majority of the vehicles. For instance, both solutions reduce the travel time for 60% of the vehicles (see values lower than 1 in Figure 8(a)), while increasing the travel time for less than 15% of them (see values greater than 1 in Figure 8(a)). Regarding the relative safety risk, both solutions reduce the safety risk for 80% of the vehicles. In this context, even with its limited knowledge about the alternatives routes taken by the other vehicles (since in SNS vehicles do not know the route taken by the all vehicles), SNS still can deal with vehicular mobility properly while improving the safety for drives and passengers. This is due to its efficient traffic reporting mechanism, which provides an accurate traffic knowledge for the vehicles to improve their mobility. It is worth noticing that, for the impaired vehicles, the increase in their travel time is lower than 5 minutes.

We learned three lessons from the results: (*i*) each vehicle only needs to know the route taken by the vehicles in its vicinity to achieve a good traffic balance, since the global information about all the vehicle routes brings minimal benefits; (*ii*) SNS is slightly less effective than EBPOP because it has a less accurate traffic condition estimation, but, the benefits of providing lower overhead and complexity time jointly with the higher system scalability overcome this generally acceptable performance loss; and (*iii*) the efficient performance of offloading the re-routing computation in each vehicle paves the way to an efficient personalized safety-based re-routing, enabling that each vehicle selects the most relevant risks that it wants to avoid.

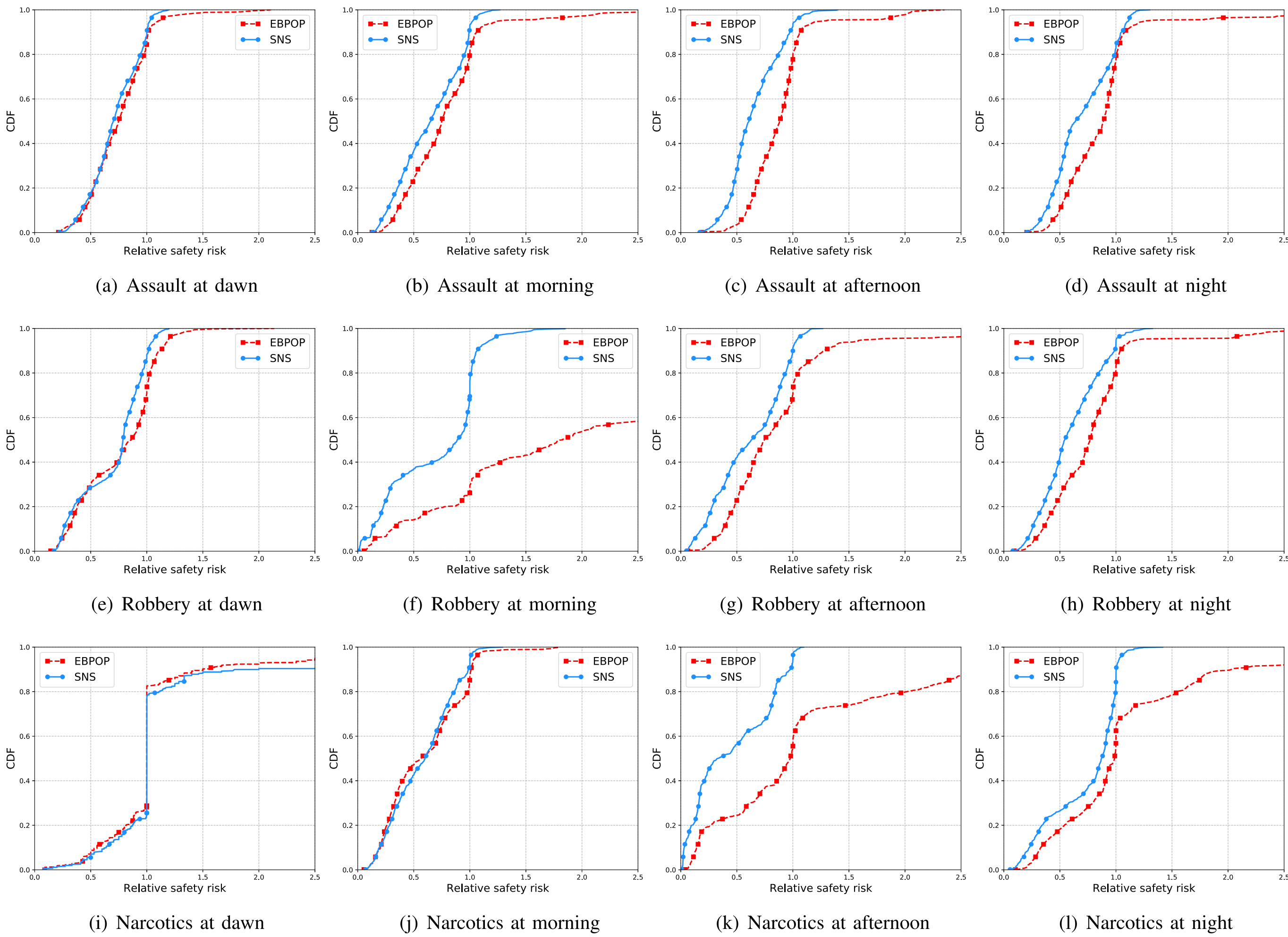
One important question is how good SNS results are when compared to state-of-the-art research in traffic re-routing. In our previous work [10], we compared EBPOP with solutions for providing fastest [19], [30], [39] and safest [35] paths as well as with solutions for improving multiple metrics including resource constrained shortest path [12] and weighted-sum [20], [27]. In summary, EBPOP outperformed those solutions in terms of mobility and safety due to its efficient context-aware re-routing algorithm which is able to provide a good traffic balance with multiple re-routing metrics. Therefore, by comparing SNS with EBPOP we can know its effectiveness against the other solutions since its performance is very close to the performance presented by EBPOP.

D. System Efficiency Analysis

With this analysis we evaluate the SNS performance considering its re-routing efficiency and compliance rate, using the following metrics:

- **Re-routing frequency**: the total number of times that each vehicle was re-routed to perform its entire trip. This metric measures the driving experience, since many route changes throughout the trip potentially decrease the quality of the driving experience.
- **Route similarity**: the similarity between the route traveled and the route initially planned by each vehicle. This metric measures the quality of the alternative routes suggested by the system; it is important to highlight that high re-routing frequency with high similarity potentially means that the alternative recommended routes were not necessary.
- **Penetration rate**: the percentage (i.e., 25%, 50%, 75%, and 100%) of vehicles that will report the traffic information and will follow the alternative computed route. This metric measures the system performance considering the scenario in which fewer vehicles use SNS. For this analysis we considered the metrics presented in Section VII-C.

Figure 9 shows the results of the re-routing frequency and route similarity metrics as a CDF. As it can be seen, the majority of vehicles are not re-routed and only 10% of the vehicles are re-routed more than once. These results show that SNS does not decrease the driving experience with frequent route changes to achieve its performance (see Figure 9(a)). Also, for those vehicles that are re-routed more than once,



(a) Assault at dawn (b) Assault at morning (c) Assault at afternoon (d) Assault at night

(e) Robbery at dawn (f) Robbery at morning (g) Robbery at afternoon (h) Robbery at night

(i) Narcotics at dawn (j) Narcotics at morning (k) Narcotics at afternoon (l) Narcotics at night

Fig. 11. Comparison of relative safety risk results for EBPOP and SNS based on different periods of the day and type of crimes during business days.

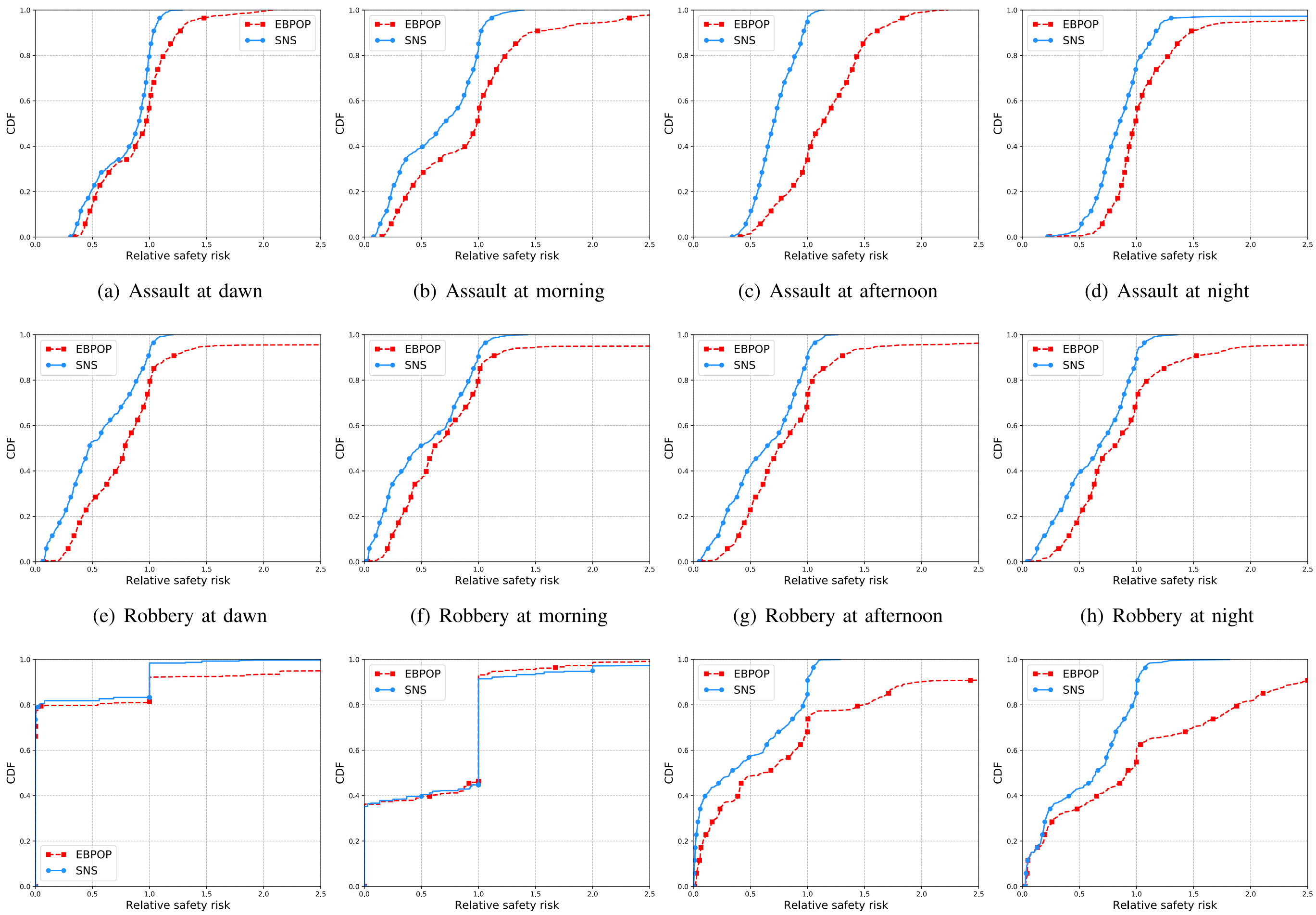
the traveled route is at most 30% similar compared to the planned route (i.e., the route planned at the beginning of the simulation). This shows that multiple route changes are only performed when it is indispensable to improve mobility and safety. Thus, multiple re-routing does not iterate throughout previously computed routes. In other words, SNS does not recommend routes to vehicles that were previously planned to themselves as new alternative routes. Finally, it is worth noticing that most vehicles are not re-routed because it is not needed (i.e., they do not pass through congested or dangerous areas). The average route similarity is close to 50% because vehicles still travel along some part of their planned route before being re-routed more than 50% of them do not change their routes at all.

Figure 10 shows the performance of SNS if only a percentage of vehicles use the system. Thus, we evaluate SNS in 25%, 50%, 75%, and 100% of the vehicles (i.e., penetration rate). As expected, the less vehicles use the system, the less messages are transmitted (see Figure 10(a)). However, even for only 25% of penetration rate, SNS still provides good traffic estimation which is higher than 90%, since a traffic report from a single vehicle in a road is enough to represent the traffic condition in that road (see Figure 10(b)). In this way, SNS can compute reliable routes for the vehicles to improve their mobility and safety of driver and passengers (see Figures 10(c) and 10(d)). Yet, the more vehicles are re-routed, the better the SNS performances are. Thus, SNS improves the mobility and safety for at least 40% of the vehicles considering all values of penetration rate, while only decreasing the mobility and safety of less than 20% of the vehicles. The inferior performance according to the penetration rate directly increases the simulation time, since vehicles spend more time to finish their trips.

In summary, the penetration results have shown that SNS is a suitable solution for dealing with traffic mobility and driver safety risks not only when some drivers do not share the traffic report, but also when they do not follow the routes recommended by the system.

E. Personalized Safety Risk With Spatiotemporal Analysis

Different drivers and passengers (i.e., vehicle) potentially have different preferences according to the safety risk provided by each crime. These preferences can vary not only according to each profile such as gender and age, but also according to the day and time [2]. Hence, based on each preference of each vehicle, the safety risk potentially changes for them, even



(a) Assault at dawn (b) Assault at morning (c) Assault at afternoon (d) Assault at night

(e) Robbery at dawn (f) Robbery at morning (g) Robbery at afternoon (h) Robbery at night

(i) Narcotics at dawn (j) Narcotics at morning (k) Narcotics at afternoon (l) Narcotics at night

Fig. 12. Comparison of relative safety risk results for EBPOP and SNS based on different periods of the day and type of crimes during weekends.

TABLE III

AVERAGE ROUTE SAFETY RISK FOR ASSAULT-RELATED CRIMES ON BUSINESS DAYS AND WEEKEND CONSIDERING DIFFERENT PERIODS OF THE DAY

	Business days				Weekend			
	Dawn	Morning	Afternoon	Night	Dawn	Morning	Afternoon	Night
No Re-routing	16.07	14.02	17.03	16.65	11.62	3.55	16.54	18.61
EBPOP	11.04	9.55	13.67	12.87	10.15	3.00	15.07	17.98
SNS	10.76	8.54	10.72	11.00	8.87	2.25	10.52	15.07

for vehicles with same origin and destination but with different preferences according to type of crime that they want to avoid.

Since SNS enables personalized re-routing and also explores the spatial and temporal information about dangerous areas, with this analysis we want to assess how these preferences and the spatiotemporal correlation of criminal activities can impact the safety of the drives and passengers. To do so, we differentiate the days into *Business days* and *Weekend* and also segmented each day into four periods named as: (*i*) Dawn, from 00:00 to 05:59; (*ii*) Morning, from 06:00 to 11:59; (*iii*) Afternoon, from 12:00 to 17:59; and (*iv*) Night, from 18:00 to 23:59. In this way, by randomly selecting a day and time we predicted the criminal density for each type of crime using the RNN and perform the vehicular traffic re-routing considering each crime preference.

TABLE IV

AVERAGE ROUTE SAFETY RISK FOR ROBBERY-RELATED CRIMES ON BUSINESS DAYS AND WEEKEND CONSIDERING DIFFERENT PERIODS OF THE DAY

	Business days				Weekend			
	Dawn	Morning	Afternoon	Night	Dawn	Morning	Afternoon	Night
No Re-routing	14.05	0.65	13.10	14.58	11.78	11.44	10.88	10.51
EBPOP	9.88	0.63	9.91	9.43	8.22	6.76	7.91	8.46
SNS	9.47	0.10	8.24	8.20	6.19	6.28	5.66	6.88

TABLE V

AVERAGE ROUTE SAFETY RISK FOR NARCOTICS-RELATED CRIMES ON BUSINESS DAYS AND WEEKEND CONSIDERING DIFFERENT PERIODS OF THE DAY

	Business days				Weekend			
	Dawn	Morning	Afternoon	Night	Dawn	Morning	Afternoon	Night
No Re-routing	0.26	10.35	3.26	4.86	0.14	0.07	4.49	3.61
EBPOP	0.25	4.78	2.81	4.58	0.04	0.06	3.35	2.97
SNS	0.24	4.54	1.31	3.19	0.01	0.05	1.73	1.72

Figure 11 shows the relative safety risk results for each period of the day for the type of crime during business days comparing SNS and EBPOP, while Figure 12 shows the same results during weekends. Also, Tables III, IV and V show the average route safety risk result (e.g., r_P) for each solution considering each type of crime. As it can be seen, SNS outperforms EBPOP for all days and periods considering the different safety risks provided by each criminal activity (see Figure 11 and Figure 12). This is the result of the preference-based re-routing employed by SNS, in which vehicles just consider the relevant risk for them, while in EBPOP the system just knows the overall risk, not considering which criminal activity affects each part of the city according to the day and time. Thus, if another criminal activity has a higher density in some area than the other criminal activities, vehicles that have preference about safety risks with lower density will be impaired, because EBPOP can guide them towards regions that they do not want, to avoid other areas with higher density, consequently increasing the safety risk when considering their preferences. However, in SNS vehicles just avoid the regions and the safety risks relevant for them, which contributes to improve their safety according their preference.

In particular, SNS reduces the average safety risk in approximately 30% when compared to EBPOP considering the safety risk provided by different criminal activities and their spatiotemporal correlation. This is result of the accurate criminal density prediction provided by the RNN and the efficiency of the cooperative re-routing algorithm offloaded in each vehicle, which enables the personalized re-routing to improve the safety of drivers and passengers based on their preferences.

VIII. Conclusion

This work proposed SNS, a non-deterministic personalized driver safety-ware vehicular re-routing system that considers the predicted safety risks and spatiotemporal information to improve the mobility and reduce safety risks to drivers and passengers. SNS is able to extract pieces of knowledge about both traffic conditions and risky areas based on vehicular mobility and criminal incidents. Thus, by measuring the safety risk of each region considering each type of crime and its spatiotemporal correlation, SNS employs an RNN to predict future safety risks concerning each criminal activity. Moreover, by offloading the re-routing computation in each vehicle, SNS enables a personalized re-routing in which each vehicle (e.g., driver) can decide which crime it wants to avoid.

Simulation results have shown that SNS reaches substantial improvements regarding system scalability, time complexity, and safety. In particular, compared to EBPOP, SNS reduces approximately 30% the average safety risk for drivers and passenger considering their preferences. Also, SNS keeps the same traffic efficiency and improves in approximately 99% the CPU time of the re-routing algorithm, which is the consequence of its accurate predictions about future safety risks jointly with the effectiveness of the cooperative personalized re-routing algorithm implemented by SNS.

It is worth noticing that the same re-routing approach can be extended to incorporate different metrics to be personalized such as time, cost, fuel consumption, CO emissions, and social features (scenery, smell, noise, etc). As future work, in addition to the different personalized metrics, we also want to use social media information and participatory sensing more dynamic and real-time sensing to identify unexpected events that can suddenly change the traffic efficiency and safety risk in some region.

References

[1] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[2] M. Amirgholy, N. Golshani, C. Schneider, E. J. Gonzales, and H. O. Gao, "An advanced traveler navigation system adapted to route choice preferences of the individual users," *Int. J. Transp. Sci. Technol.*, vol. 6, no. 4, pp. 240–254, 2017.

[3] J. Ballesteros, M. Rahman, B. Carbunar, and N. Rishe, "Safe cities. A participatory sensing approach," in *Proc. IEEE 37th Conf. Local Comput. Netw. (LCN)*, Oct. 2012, pp. 626–634.

[4] J. H. Banks, *Introduction to Transportation Engineering*. New York, NY, USA: McGraw-Hill, 2002.

[5] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of urban MObility: An overview," in *Proc. Int. Conf. Adv. Syst. Simulation*, 2011, pp. 63–68.

[6] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G network slicing for vehicle-to-everything services," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 38–45, Dec. 2017.

[7] Z. Cao, H. Guo, and J. Zhang, "A multiagent-based approach for vehicle routing by considering both arriving on time and total travel time," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 3, p. 25, Dec. 2017.

[8] Z. Cao, S. Jiang, J. Zhang, and H. Guo, "A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1958–1973, Jul. 2017.

[9] *Waze App Directions Take Woman to Wrong Brazil Address, Where She is Killed*, CNN, 2015. [Online]. Available: https://edition.cnn.com/2015/10/05/americas/brazil-wrong-directions-death/index.html

[10] A. M. De Souza, T. Braun, and L. Villas, "Efficient context-aware vehicular traffic re-routing based on Pareto-optimality: A safe-fast use case," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2905–2910.

[11] A. M. de Souza, C. A. Brennand, R. S. Yokoyama, E. A. Donato, E. R. Madeira, and L. A. Villas, "Traffic management systems: A classification, review, challenges, and future perspectives," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 4, 2017, Art. no. 1550147716683612.

[12] A. M. de Souza, L. L. C. Pedrosa, L. C. Botega, and L. Villas, "Itssafe: An intelligent transportation system for improving safety and traffic efficiency," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–7.

[13] A. M. de Souza and L. A. Villas, "A fully-distributed traffic management system to improve the overall traffic efficiency," in *Proc. 19th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst. (MSWiM)*, New York, NY, USA, 2016, pp. 19–26.

[14] A. M. de Souza *et al.*, "ICARUS: Improvement of traffic condition through an alerting and re-routing system," *Comput. Netw.*, vol. 110, pp. 118–132, Dec. 2016.

[15] A. M. de Souza, R. S. Yokoyama, L. C. Botega, R. I. Meneguette, and L. A. Villas, "Scorpion: A solution using cooperative rerouting to prevent congestion and improve traffic condition," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervasive Intell. Comput.*, Oct. 2015, pp. 497–503.

[16] A. M. de Souza, R. S. Yokoyama, G. Maia, A. Loureiro, and L. Villas, "Real-time path planning to prevent traffic jam through an intelligent transportation system," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 726–731.

[17] S. Djahel, R. Doolan, G.-M. Muntean, and J. Murphy, "A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 125–151, Mar. 2015.

[18] P. Dong, T. Zheng, S. Yu, H. Zhang, and X. Yan, "Enhancing vehicular communication using 5G-enabled smart collaborative networking," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 72–79, Dec. 2017.

[19] R. Doolan and G.-M. Muntean, "EcoTrec—A novel VANET-based approach to reducing vehicle emissions," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 608–620, Mar. 2017.

[20] E. Galbrun, K. Pelechrinis, and E. Terzi, "Urban navigation beyond shortest route," *Inf. Syst.*, vol. 57, pp. 160–171, Apr. 2016.

[21] H. Joksch, "The shortest route problem with constraints," *J. Math. Anal. Appl.*, vol. 14, no. 2, p. 191– 197, 1966.

[22] S.-W. Kim and W. Liu, "Cooperative autonomous driving: A mirror neuron inspired intention awareness and cooperative perception approach," *IEEE Intell. Transp. Syst. Mag.*, vol. 8, no. 3, pp. 23–32, Jul. 2016.

[23] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, p. 13, Sep. 2015.

[24] J. Liu *et al.*, "High-efficiency urban traffic management in context-aware computing and 5G communication," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 34–40, Jan. 2017.

[25] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[26] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidisciplinary Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004.

[27] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: New insights," *Structural Multidisciplinary Optim.*, vol. 41, no. 6, pp. 853–862, 2010.

[28] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of recurrent neural networks for probabilistic modeling of driver behavior," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1289–1298, May 2017.

[29] M. Online. (2016). *Dash Cam Captures the Terrifying Moment Waze Smartphone App Directs a Driver Into a Gunfight in Boston*. [Online]. Available: https://www.dailymail.co.uk/news/article-3717925/Dash-cam-captures-terrifying-moment-Waze-smartphone-app-directs-driver-GUNFIGHT-Boston.html

[30] J. S. Pan, I. S. Popa, and C. Borcea, "Divert: A distributed vehicular traffic re-routing system for congestion avoidance," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 58–72, Jan. 2017.

[31] L. G. Papaleondiou and M. D. Dikaiakos, "TrafficModeler: A graphical tool for programming microscopic traffic simulators through high-level abstractions," in *Proc. VTC Spring IEEE 69th Veh. Technol. Conf.*, Apr. 2009, pp. 1–5.

[32] Z. Peng, S. Gao, Z. Li, B. Xiao, and Y. Qian, "Vehicle safety improvement through deep learning and mobile sensing," *IEEE Netw.*, vol. 32, no. 4, pp. 28–33, Jul./Aug. 2018.

[33] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, Aug. 2018.

[34] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York, NY, USA: Wiley, 1992.

[35] S. Shah, F. Bao, C.-T. Lu, and I.-R. Chen, "Crowdsafe: Crowd sourcing of crime incidents and safe routing on mobile devices," in *Proc. 19th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, New York, NY, USA, 2011, pp. 521–524.

[36] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2010.

[37] A.-E. Taha and N. AbuAli, "Route planning considerations for autonomous vehicles," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 78–84, Oct. 2018.

[38] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. Int. Conf. Simulation Tools Techn. Commun., Netw. Syst. Workshops (Simutools)*, 2008, pp. 1–10.

[39] S. Wang, S. Djahel, Z. Zhang, and J. McManis, "Next road rerouting: A multiagent system for mitigating unexpected urban traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2888–2899, Oct. 2016.

[40] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[41] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu, "Machine learning for vehicular networks: Recent advances and application examples," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 94–101, Jun. 2018.

Allan M. de Souza received the B.Sc. degree in computer science from the University Eurípdes de Marilia (UNIVEM), Brazil, in 2013, and the M.Sc. degree in computer science from the University of Campinas (UNICAMP), Brazil, in 2016, where he is currently pursuing the Ph.D. degree. He is now under a joint-supervision degree program with the University of Bern, Switzerland. His research interests are in the field of data dissemination, congestion detection, and congestion control and re-routing in VANETs and traffic management systems.

Torsten Braun received the Ph.D. degree from the University of Karlsruhe, Germany, in 1993. From 1994 to 1995, he was a Guest Scientist with the INRIA Sophia-Antipolis, France. From 1995 to 1997, he was with the IBM European Networking Centre, Heidelberg, Germany, as a Project Leader and a Senior Consultant. He was the Director of the Institute of Computer Science and Applied Mathematics, University of Bern. Since 1998, he has been a Full Professor of computer science with the University of Bern. He has been the Vice President of the SWITCH (Swiss research and education network provider) Foundation.

Leonardo C. Botega received the Ph.D. degree in computer science from the Federal University of São Carlos (UFSCar), with a postdoctoral internship at the University of São Paulo (USP). He is a Professor of computer science and information systems with the University Center Euripides of Marília (UNIVEM). He is currently a member of the Faculty of Postgraduate Program in Information Science with São Paulo State University "Júlio de Mesquita Filho" (UNESP). He is a Researcher in critical data management projects with experience in critical decision making systems, ontology engineering, and cognitive computing. He is the Editor-in-Chief of *Journal on Advances in Theoretical and Applied Informatics* (JADI).

Leandro A. Villas received the Ph.D. degree in computer science from the Federal University of Minas Gerais, Brazil, with a partial doctoral fellowship at University of Ottawa, Canada, in 2012. He is currently an Associate Professor of computer science with the University of Campinas (UNICAMP), Brazil. His research interests include localization and synchronization algorithms, distributed algorithms, and wireless ad hoc, vehicular, and sensor networks. He is an author of several articles in the different areas of his research interests.

Antonio A. F. Loureiro received the B.Sc. and M.Sc. degrees in computer science from the Federal University of Minas Gerais (UFMG), Brazil, and the Ph.D. degree in computer science from the University of British Columbia, Canada. He is currently a Full Professor of computer science with UFMG, where he leads the research group in wireless sensor networks. His main research areas are wireless sensor networks, mobile computing, and distributed algorithms. In the last ten years, he has published regularly in international conferences and journals related to those areas, and also presented tutorials at international conferences.