Applying Network Kernel Density Estimation (NKDE) and Temporal Network Kernel Estimation (TNKDE) for Generating Safer Routes

Antonios Karatzoglou
Microsoft Corporation
Maps & Geospatial Services
Mountain View, CA, USA
antonios.karatzoglou@microsoft.com

ABSTRACT

Drivers nowadays expect increasingly more from their navigation systems. For this reason, navigation software providers move towards offering more sophisticated and context-aware solutions. At the same time, the global report from WHO on road safety shows increasingly alerting number when it comes to modern traffic. Thus, reducing the likelihood of road traffic accidents is an important and timely topic. In this paper, we explore the use of the *Network Kernel Density Estimation (NKDE)* and the *Temporal Network Kernel Density Estimation (TNKDE)* functions, which, as opposed to the plain KDE methods, utilize road network graph data, as basis for identifying and generating safer routes. We evaluate our approach using a traffic accident dataset for the city of San Francisco against a vanilla two-dimensional Kernel Density Estimation and a K-Means clustering method. Furthermore, we investigate the role of the degree of severity of the accidents and its impact on the overall result. Our work shows that (T)NKDE-based methods can be used for identifying safer routes in a road network without compromising on the overall trip distance and duration.

CCS CONCEPTS

• **Applied computing** → **Transportation**; Multi-criterion optimization and decision-making; Consumer products; • **Theory of computation** → **Probabilistic computation**.

KEYWORDS

Navigation, Routing, Safer Routes, Traffic Accidents, Kernel Density Estimation, Road Network, Location Based Services

ACM Reference Format:
Antonios Karatzoglou. 2022. Applying Network Kernel Density Estimation (NKDE) and Temporal Network Kernel Estimation (TNKDE) for Generating Safer Routes. In *The 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS '22), November 1, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3557991.3567782

IWCTS '22, November 1, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9539-7/22/11…$15.00
https://doi.org/10.1145/3557991.3567782

1 INTRODUCTION

Based on WHO's *Global status report on road safety* [24], road traffic accidents have become the 8^{th} leading cause of death for people of all age groups and the 1^{st} cause of death for children and young adults aged 5-29 years old. Overall, we're facing a total number of more than 1.35 million traffic deaths each year, an unacceptably high number for the modern society we leave in. It is self-explanatory that every effort aiming at lowering this number as much as possible and as fast as possible is of significant importance. This can be achieved in various ways. For instance by launching campaigns and enforcing legislation measures on WHO's identified key risk factors such as on speed limits, drug- and drink-driving as well as on helmet, seat-belt and child restraint usage in a more unified and consistent way around the globe. A large number of traffic incidents is attributed to the road infrastructure and road characteristics [25, 28]. Existing road safety rating systems help monitor the road quality and identify high risk roads and rank them by the crash likelihood and severity. On the one hand, the resulting insights can be used to improve the road design, which can have a massive positive effect on the traffic accident statistics. On the other hand, the results can be used for managing and redirecting the traffic in order to avoid dangerous hotspots. Intelligent traffic management and re-routing is doubly important since the traffic volume (as a ratio to the road capacity) is an important risk factor itself [6, 11, 41]. At the same time, redirecting the traffic at a single-user level can be equally important. A fact that resonates with the recent hype in the research literature around risk-aware routing and direction services that take safety explicitly into account.

In its first part, this work goes through the literature and highlights the most related recent work (Section 2). The second part introduces, describes and evaluates our approach for deriving the safest routes as well as its impact on travel time, travel distance and carbon emissions against a number of conventional methods (Section 3). Finally, in our conclusion and future work section we summarize our findings and discuss potential future work items (Section 4).

2 RELATED WORK

Finding and recommending the safest routes in a pedestrian, cyclist or driver navigation scenario, is a relatively new but increasingly discussed topic in the research community. We can distinguish two types of safe route recommendation systems. The first type includes systems that are closely interlinked with the underlying routing algorithm and route generation process. That is, systems

that influence the typical edge selection process during the route composition phase. The second type represents approaches that are build on top of the routing engine as an additional layer and help rank and highlight the returned routes with respect to the associated risk or safety. In addition, *safe routing* titled papers address primarily two major risk root causes while navigating through a road network, namely *traffic accidents* and *crime*, whereas the latter concerns mostly pedestrian navigation scenarios with some of them relying on crowd-sourced data or on mining social media as seen in [16, 29] and [8, 18] respectively. Despite the different risk reason, crime-related papers often share similar strategies for avoiding the risk areas along a user's route and thus, making many of them relevant to our work and worth mentioning. This section references the ones that are most closely related to our approach.

A recent study suggests that eliminating left turns can significantly reduce the accidents risk [5]. Thus, finding a safer route can be as easy as picking the route that has fewer left turns! But before calling it for the day, let's take a look how this can be further improved by taking a more data-driven approach. With *CrowdPath* for instance, Hendawi et al. present a framework for a crowd-based routing engine [15]. Their framework operates upon the result of a set of potential route providers and uses volunteers' data to generate routes that reflect the choice of local drivers. At the same time, their time-dependent architecture supports taking into account a number of risk factors such as crime and accident risk, although they rather include the implementation and evaluation of this feature into their later work *PreGo* [12–14]. PreGo constitutes an evolved and more mature version of CrowdPath in which users are allowed to both provide their preferences with respect to travel time, distance, risk and number of services that can be found along a route as well as to contribute with their own data (e.g., report an accident at a certain location). Apart from PreGo itself, their work in [12] includes an interesting user study about the users' route selection criteria. Their study shows that the drivers' preferences with respect to a route's travel time, safety, "scenicness" and utility varies massively based on the trip scenario (commute vs. shopping vs. vacation trip), on the time of the day and the day of the week, as well as on the gender of the driver. Both CrowdPath and PreGo seem not to take the distance between the route segments and the risk source into the risk factor estimation process. Aljubayrin et al. on the other hand, describe an algorithm that does exactly that [3]. Their work introduces the notion of so-called *safe zones* and *preferred zoned* and their algorithm navigates users through a route that minimizes the distance travelled outside of them. A safe zone might be a village in a desert or a neighborhood near a police station and their location is defined by the authors, as opposed to our data-driven approach that generates safe areas via spatial point pattern analysis. Makarova et al. focus on safe routes specifically for pedestrians and cyclists in cities as a way to reduce the number of road accident victims [20]. Their work on pedestrians and cyclists aligns in a certain way with the global long-term efforts, programs and case studies on promoting and identifying *Safe Routes to School (SRTS)* for children (and not only) using their bike or going on foot [22, 32, 37]. Krumm and Horvitz introduce in their work a probabilistic method to calculate the crash risk on any road in a given road network as a function of the traffic volume, a set of road attributes and a set of environmental conditions [19]. The inferred risks based on the identified probability distributions flow as costs into a Dijsktra-based route planner to generate the safest routes. In addition, their work studies the trade-offs between safety and the route's total distance or travel time. In contrast to our work, their work relies on parametric statistics and does not take into account the severity of the traffic incidents, though it is mentioned in their future work section. Soni et al.'s method [31] on the other hand takes the incident's severity degree into account. Their work considers both crime and accidents to generate risk scores along a route, while assigned author-defined weights let the corresponding severity flow into the overall risk score. Moreover, Soni et al. make use of a clustering method to infer high risk areas from the available data. Clustering has been proven to be an adequate data mining method as also seen in [1]. In particular, Soni et al.'s method applies a nested clustering algorithm in combination with a distance-aware kNN regression (as opposed to our KDE function) to describe a set of routes by their safety. The kNN regression, with k an author-defined system parameter, let's the distance between a route segment and the identified risk clusters play a part in the risk assessment, however, their work doesn't seem to address time dependency as we do in this paper. Galbrun et al.'s work focuses on reducing the crimes rather than the traffic accident risks during an urban navigation scenario [9]. A kernel function is used for computing the crime probability on each route edge, which in turn is assigned as an additional cost to a Dijkstra-based routing algorithm. At the same time, the authors lay particular weight on finding a computationally optimized algorithm and they study various strategies such as early stopping and path set pruning for this purpose. Finally, their work highlights the importance of taking into account the population density and the temporal dimension. However, both aren't handled in their paper and are referred to as future work. Furthermore, they use the *simpler* planar KDE as opposed to our network-based KDE. Asawa et al.'s paper is also focusing on determining the likelihood of a criminal offense along a certain route [4] . Like some of the previously mentioned work, they too use clustering to identify criminal hotspots. A predetermined radius is used to define the nearest clusters to the route, while the number of the crimes as well as the type of the crime within each cluster affects the overall risk score. Unlike aforementioned work, their method lets user profile information (age and gender) flow into the risk assessment process through a Bayesian Network that computes the likelihood of an individual becoming a crime victim at a given time. In order to achieve this, users are being asked to provide the corresponding information when signing up into their tool. Finally, a special group of papers, such as [34, 42] to name but a few, address the problem of transportation of hazardous materials and how to find the safest route for both the driver but also the people that reside near the route in terms of exposure and release of dangerous material .

The work presented in this paper attempts to combine in a joint approach the individual missing elements from the aforementioned related work and explores a static as well as a dynamic, time-dependent Network-based KDE as a means for identifying less accident-prone paths on a road network, while keeping an eye on the severity degree of the traffic incidents in the available data at the same time.

3 NETWORK KERNEL DENSITY ESTIMATES (NKDE) FOR SAFER ROUTES

This section goes through the main spatial pattern analysis methods that were applied on an US traffic accident dataset on our way to ultimately exploring and defining the *Network Kernel Density Estimate (NKDE)* to be the basis for our safe routes layer. Based on the literature, we focused primarily on clustering and kernel density based methods.

The US Accident dataset [21] that we used for our experiments spans over many years, namely from 2016 to 2021[1] and comprises about 2.8 million accident records. Apart from expected seasonal variances and despite the Corona-related pandemic during the last 2 years of the dataset, the data show a certain consistency, which we assume to be sufficient for extracting meaningful insights out of the data. So, apart from basic checking through the data for empty values or unreasonable outliers, no other data have been removed during our preprocessing step. In this work, we selected the region of San Francisco city to explore and evaluate our method. It could be noted here, that traffic data like these become increasingly available from official municipal and other government authorities around the world. At the same time, similarly useful data, such as critical driving behaviour and patterns, can be successfully inferred and generated from vehicle or driver datasets, as already seen in the literature [2, 7, 17, 36].

3.1 K-Means Clustering Analysis

Clustering methods represent a simple and intuitive unsupervised way for identifying spatial patters in geospatial data. K-Means stands out as a computationally efficient method (with a linear complexity $O(n)$), frequently used in scenarios like ours, as we have seen previously in the Related Work section. We used K-Means to get a first impression of the available traffic incident data and evaluate whether this could be useful for our safe routes approach.

K-Means needs a predefined number of clusters c. We tested various values and made our selection ($c = 4$) based on the *Elbow* method (see Fig. 1).

Figure 1: Applying the *Elbow* method to define the optimal number of clusters.

If we apply K-Means with $c = 4$ cluster centroids we get the image in Fig. 2. Fig. 2 shows the traffic incidents projected on top of the road network of San Francisco colored based on their cluster.

[1]Stand 08/26/2022

The yellow circles represent the 4 clusters, that is, the location of their centroids, whereas their size is analog to the number of incidents belonging to each cluster. We can see that the clustering approach provides certain insights into the 4 most problematic areas in the city, the areas around the Golden Gate and the Bay bridge, as well as around the high traffic highways 101, 280 and 80. However, using these clusters for identifying low-risk roads in terms of accidents by a distance/radius-based approach as seen in crime-related safe route literature seems rather like a long shot. Even if for some cases looks relatively reasonable, like for the case of the largest cluster on the north-west side of the city, where the traffic incidents are spread along many roads in a somewhat circular way. For all the other cases, a distance-based approach from the centroid would naively assign high risk values to safe roads, just because being close to a a road that is indeed dangerous. This might make more sense in a pedestrian navigation scenario for avoiding high-risk crime regions, but less for our driving scenario. Our high-risk areas need to be defined in a much higher precision than that. This is where the Kernel Density Estimation might be helpful.

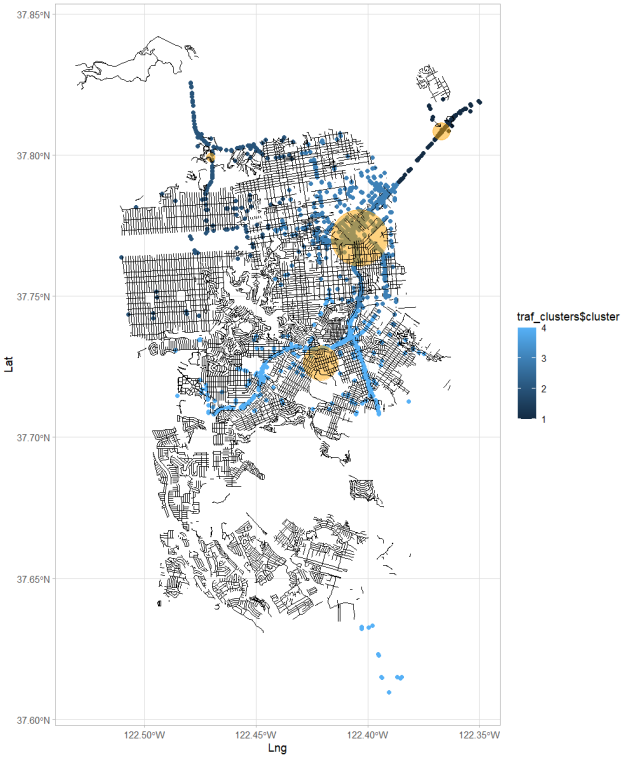


Figure 2: Traffic accident K-Means clusters in San Francisco from 2016 to 2021. The size of the (yellow) cluster centroid is analog to the number of accidents belonging to that cluster.

3.2 Kernel Density Estimation (KDE) Analysis

Kernel Density Estimation (KDE) is a sample- and kernel-based non-parametric statistical method for estimating the probability density function of a random variable, which in turn reflects its

occurrence probability. KDE is a handy tool that relieves as from the need to make assumptions about the distribution of our data and the corresponding statistical tests to confirm (or not) our hypothesis. It uses *kernels*, which in the 2-dimensional case (i.e., in the case of geospatial points) are typically smooth, circular, bell-alike curved surfaces, that are slid and fitted along a grid area defined by our data sample in order to identify low- and high-density sample regions.

The general formula of a planar 2-dimensional Kernel Density Estimation function for a location point $p(x, y)$ is given by Eq. 1 [38]:

$$Density2D = \sum_{i=1}^{n} \frac{1}{\pi r^2} \cdot k \cdot \frac{d_i}{r} \tag{1}$$

where i are the location points in the available data sample that lie within the radius r (aka *bandwidth*) from point p, k is the *Kernel* function that returns a weight for the point i at distance d_i based on the ratio between d_i and r and d_i is the distance between the point i and p.

Equation 2 describes a version of a 2-dimensional Kernel Density function that uses the frequently used *Quartic* function as its kernel to estimate the density of a location point $p(x, y)$ as found in [30] and implemented in various GIS tools[2].

$$Density2D_{Quart}(p) = \frac{1}{r^2} \cdot \sum_{i=1}^{n} \left(\frac{3}{\pi} \cdot pop_i \left(1 - \left(\frac{d_i}{r} \right)^2 \right)^2 \right) \tag{2}$$

where i are the location points of interest in the available data sample that lie within the radius r (aka *bandwidth*) from point p, pop_i is an optional weight parameter of the point i called *population field*, and d_i is the distance between the points i and p. Beside choosing an appropriate kernel, the selection of the bandwidth r is important for the performance of the KDE and an optimal bandwidth value is often found by making use of thumb rules that rely on statistical metrics such as the mean and the standard deviation applied on available sample points. Narrow bandwidths help unveil local phenomena, while a wider bandwidth rather highlights *hospots* in large areas.

We experimented with various kernels. Fig. 3 shows the results of 2-dimensional kernel density estimation function that uses an axis-aligned bivariate Gaussian kernel evaluated on a square grid. We used the R library *MASS* and the default bandwidth is based on a thumb rule found in [35] that in turn relies on the 25^th and the 75^th quantile of our sample. It can be clearly seen that KDE helps define the high accident risk zones much more precisely than K-Means in the previous section. It covers the 4 high risk areas from K-Means in a more compact and decisive manner and leaves a big part of the *irrelevant* safer roads out of its range. It seems reasonable and feasible to use the resulting traffic incident densities assigned as weights to the set of streets lying underneath the KDE to help a routing algorithm avoid those and find safer alternative routes. However, Fig. 3 also shows some bandwidth-coverage trade-off limitations of the KDE approach when it occasionally misses to highlight a few significant accident-prone street segments as well as a small portion of significant scattered points that lie away from

[2]R Sp, ArcGIS Pro, ..

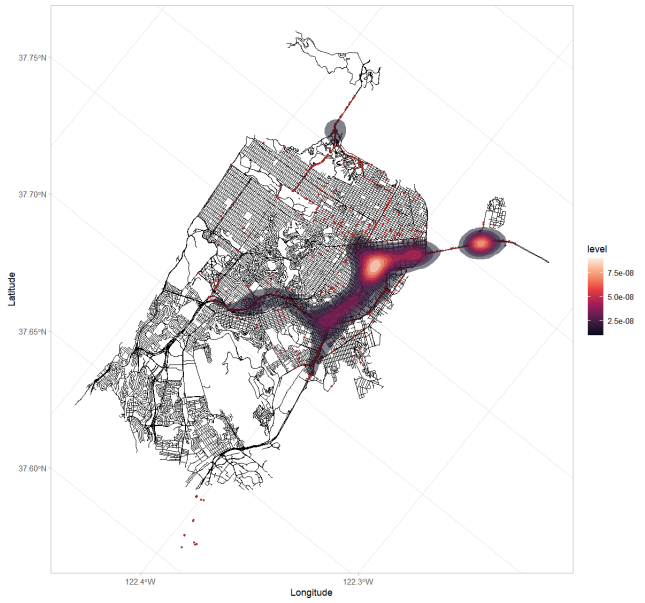


Figure 3: KDE traffic accident density distribution projected on the San Francisco road network.

the hotspots. At the same time, even with this compact hotspot representation, KDE indicates a number of streets as risky just because these are geometrically closer to the actual risky roads, similar to the aforementioned K-Means distance-based approach. Although in a much smaller degree, it is still not optimal and there is space for improvement.

Aiming at tackling the missing accident areas, we experimented with a *rastered* version of KDE. Fig. 4 shows an attempt to map Gaussian KDE density values to $2km \times 2km$ grid cells projected on the road network. This rastered approach makes density variations

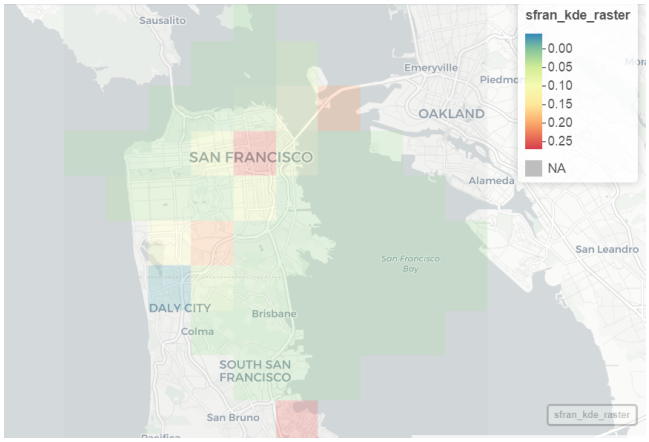


Figure 4: Rastered KDE traffic accident density distribution projected on the San Francisco road network.

across the city more visible and the resulting cells can then be used

for the path costs generation in the routing algorithm. By tuning the grid resolution we are able to tune the level of precision, eventually reaching the level seen in Fig. 3, which would bring us to a similar bandwidth-coverage trade-off issue.

All in all, KDE showed promising results, but it still led to artifacts were safe roads are considered to be of a high risk. It lacks the precision that is required for a traffic scenario such as ours. Mostly because like K-Means it doesn't take the underlying road network into account. Traffic bound to road networks doesn't move to all 360° directions, that is, it isn't *isotropic*, as is the assumption for above techniques that use the Euclidian distance. Here is where the Network KDE comes in place (Section 3.3).

3.3 Network Kernel Density Estimates (NKDE)

While the common 2-dimensional KDE estimates densities over area units, the *Network Kernel Density Estimation (NKDE)* technique estimates densities over a linear unit via a *lixelization* process [38]. This enables it to use a network space, such as a road network, as the data point event context for the events we want to estimate the densities, e.g., the traffic incidents in our case. Eq. 3 shows the Network KDE function for a location point p:

$$NKDE(p) = \sum_{i=1}^{n} \frac{1}{r} \cdot k \cdot \frac{d_i}{r} \tag{3}$$

where i are the location points of interest (e.g., the traffic incidents) in the available data sample that lie within the radius r (aka *bandwidth*) from point p, k is the *Kernel* function and d_i is the distance between the point i and p. It can be seen that by eliminating the πr^2 term from the general KDE (Eq. 1) and replacing it with r, this function now estimates the density over a linear unit, e.g., a road segment. Moreover, both the bandwidth selection process and the kernel function use the network distance (instead of the Euclidian line of sight distance). This is done by integrating a shortest path calculation step into the NKDE algorithm. As with the conventional KDE, the choice of the Kernel and the value of the bandwidth have a significant impact on the result. On top of that comes the search for the optimal length of the so called *lixels* that NKDE generates in a preprocess step and uses to sample the underlying the road network. Since the first NKDE paper in 2008, a number of NKDE variations have been proposed that try to improve the limitations of the first paper, such as overestimation of densities at intersections and artifacts caused by network loops smaller than the bandwidth as found in [23, 33], to name but a few. However, it should be noted that the improved algorithms often come with the cost of a higher complexity and computational time. For our experiments we used the *spNetwork* library in R that includes 3 different NKDE implementations [10].

After experimenting with various kernels, bandwidth values and lixel sizes, we reached to the result shown in Fig. 5 by using a Quartic kernel with a bandwidth of 300m and a lixel size of 200m with a minimum cut-off distance of 50m. Regarding the network data used in our model, We queried our San Francsico road network

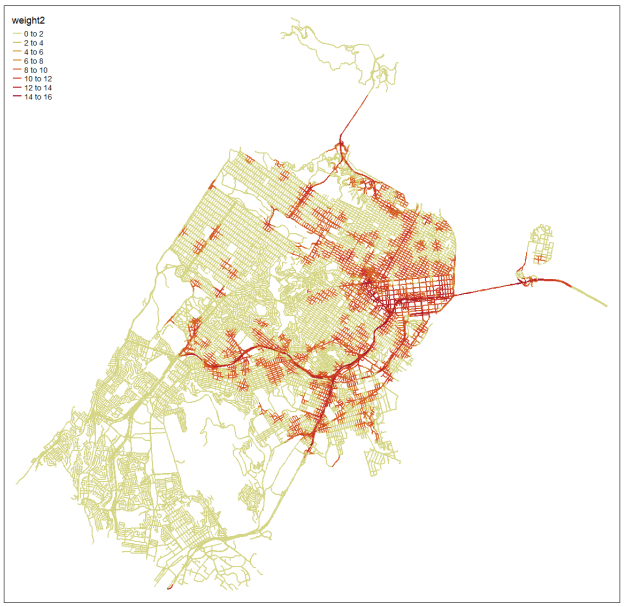
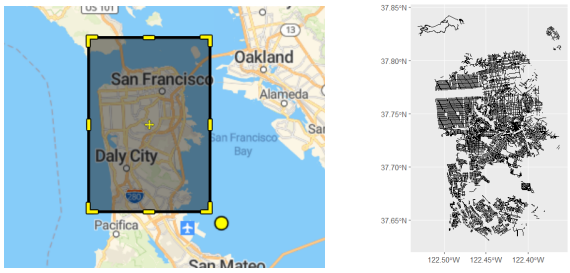


Figure 5: NKDE traffic accident density distribution projected on the San Francisco road network. (Note: The density values were scaled up for a clearer visualization)

graph from OSM[3] via the overpass API[4][5] by providing the bounding box found in Fig. 6.



(a) (b)

Figure 6: (a) Bounding box over the city of San Francisco defining our area of interest. (b) Extracted OSM road network.

We can see from Fig. 5 that NKDE gives us a very precise representation of the accident-prone streets in the road network of San Francisco, which in turn reflects the likelihood of a traffic accident in those areas. The precision is by far more superior to the one provided by the standard KDE and fits much better to our goal, namely, to navigate the user through safer routes.

Now that we have the network density matrix generated, we can use its values to assign weights to the edges of our road network graph and start deriving our safer routes. However, if we used the

[3]https://www.openstreetmap.org/
[4]https://wiki.openstreetmap.org/wiki/Overpass_API
[5]http://overpass-api.de/

generated accident risk-specific densities as sole weights in a path finding algorithm such as Dijkstra we would end up with a certain number of rather extreme results. Fig. 7 shows such an extreme example route. The reason is obviously because of the routing

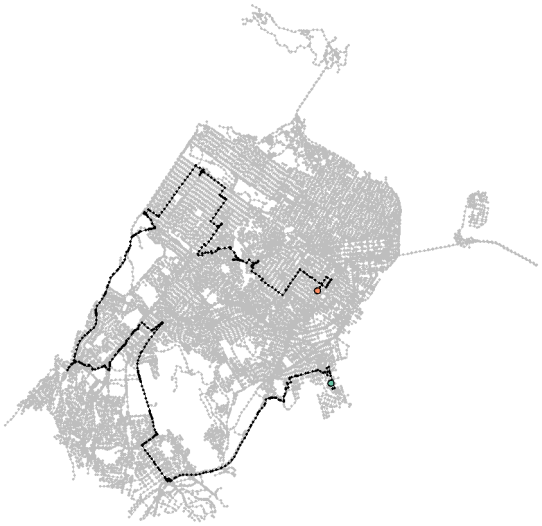


Figure 7: Extreme example of an NDKE-based safe route from an origin (green) to a destination point (orange) that ignores the length of the route.

algorithm trying to avoid all the risky streets as shown in Fig. 5 without exception and without taking the route's total distance into account. Fig. 8 shows the distance-only weight distribution of the road network edges for the sake of comparison.

A better model would allow both the degree of safety and the length (or the duration) of the route flow into the route selection process. We decided to explore this multi-objective optimization problem by defining the following weighted linear relation for computing the total combined weight (i.e., cost) of an edge in the network:

$$Cost_{Total}(egde) = \alpha \cdot Cost_{Length} + \beta \cdot Density_{NKDE} \quad (4)$$

where α and β are weight coefficients balancing the trade-off between the (normalized) edge's NKDE density value and its length. These could be either manually adjusted by the user or automatically inferred based on the user's habits and preferences. For normalizing the two types of weights we applied Min-Max normalization. We evaluated the following 3 combined $distance/safety$ scenarios against the fastest route in terms of total distance, travel duration and environmental impact: 25/75, 50/50, and 75/25. The evaluation was done on a sample of 200 random origin-destination points and we used the Dijkstra algorithm to find our optimal paths.

In order to evaluate the environmental impact, we make the naive assumption that the user drives a typical modern medium-sized

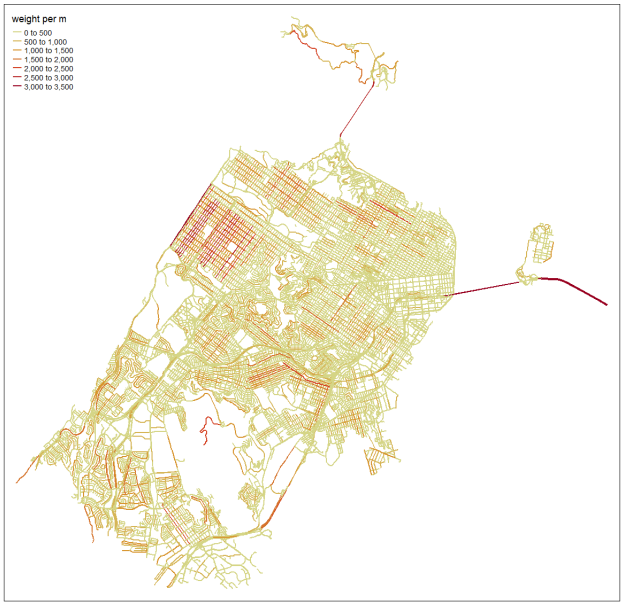


Figure 8: Distance-based edge weight distribution projected of our San Francisco road network graph.

gasoline vehicle and apply the distance-based formula found in the 2018 *Greenhouse Gas Emissions from a Typical Passenger Vehicle*[6] document written by the U.S. Environmental Protection Agency (EPA), which is given by Eq. 5:

$$\begin{aligned} CO2\ Emissions\ per\ mile &= \frac{CO2\ per\ gallon}{Miles\ per\ gallon} \\ &= \frac{8,887}{22.0} = 404\ [g] \end{aligned} \quad (5)$$

This equation is rather simplistic and does not consider further significant factors like the type and size of the vehicle, the driving behavior of the driver, the elevation changes and other road properties, the weather and similar context information. Nevertheless, it allows us a rough approximation that can be indicative towards the true values. For calculating the travel times, we assumed that the user is driving in a constant speed equal to the speed limit and we divided the edge distances by the speed to get the respective traverse times. Keeping the travel times short is important because longer travel times may lead to driver fatigue and raise the accident risk [26, 39, 40].

The results of our evaluation can be found in Fig. 9. As expected, we can see that the 100% *safe* routes stand out by showing the longest overall travel distances and duration, as well as the highest carbon emissions. This relatively large difference comes from extreme routes like the one seen in Fig. 7. Things change and become much smoother once the distance is included in the weight equation. The trend remains, once again as expected, clearly descending, where all three metrics decrease the more weight we lay on the distance and favouring in this way the shorter paths in our road

[6] https://www.epa.gov/greenvehicles/greenhouse-gas-emissions-typical-passenger-vehicle

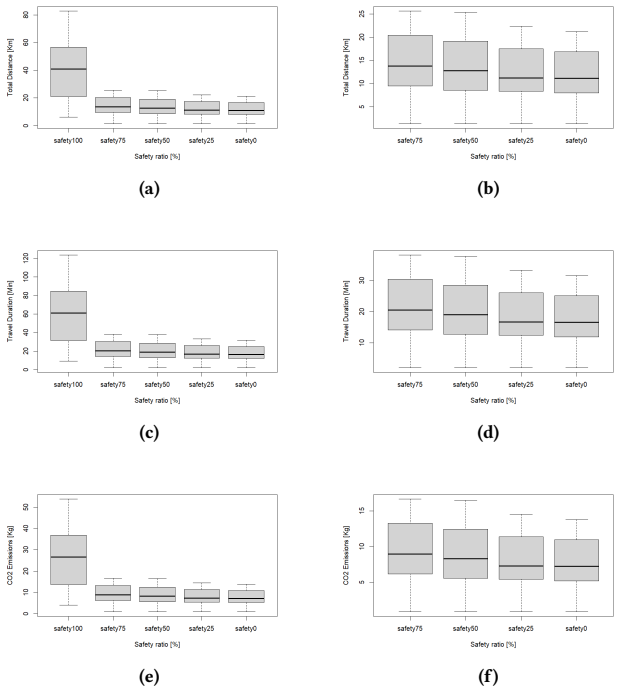


Figure 9: Evaluation results with respect to: Total travelled distance (a,b), Total travel time (c,d) and CO2 impact (e,f) by $distance/safety$ weight ratio with $safety0$ meaning distance-only generated routes (i.e., the shortest paths) and $safety100$ NKDE density-only routes. The images on the right (b,d,f) show the 4 rightmost of the 5 box plots found in the corresponding images on the left and help to better observe the descending trend.

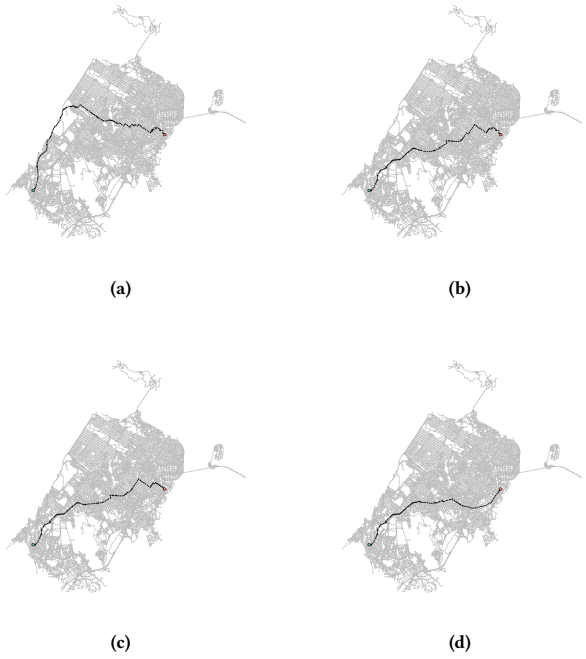


Figure 10: Comparison of the distance-safety combined weight routes for an example origin-destination point, going from the safest to the fastest route: (a) 25/75**, (b)** 50/50**, (c)** 75/25 **(d)** 100/0 **[**$distance/safety$**]** $weight\ ratio$**.**

network. Interestingly, the times and the distances are not that different when using a combined cost approach compared to the fastest route. For instance, our evaluation shows that the median difference in travel time between the 50% safe routes ($safety50$) and the fastest routes ($safety0$) is only $2.49min$! And the corresponding difference in terms of travel distance is just $1.67km$. Hence, our findings show that the NKDE-based traffic incident risk analysis can indeed be used together with the distance (and eventually other types of cost) to provide reasonable, safer, not too long and not too environmentally harmful routes. Fig. 10 underpins our findings in a visual self-explaining way for a sample route.

The US traffic accident dataset that we use for our evaluation categorizes the traffic accidents based on their severity with 1 indicating the least impact on the traffic and 4 a significant impact on the traffic (e.g., long delays). Letting the degree of severity flow into our safe route generation process could help fine-tune our suggested routes and at the same time make them more robust against the eventuality of long delays. For this reason, we explored the application of NKDE on separate single severity degree data only. In addition, we further explored a weighted form of NKDE, using the severity degree as weights and we compared the results with the initial NKDE that uses all the data. This comparison is presented in Fig. 11.

What stands out is that the traffic incidents that contribute most to the largest part of the network's hotspots (e.g., the 2 bridges and the arterial roads) are of severity 2 followed by the ones of severity 3 (Fig. 11(b,c). Severity 4 incidents are less spread and build compact, though extreme high-risk density regions, while severity 1 can be almost neglected. The weighted NKDE illustrated in Fig. 11(e) seems to reflect a reasonably balanced view on the risk of traffic incidents and presents itself slightly more compact than the initial NKDE seen in Fig. 11(f). A method that takes the incident's severity into account would rely on a group of severity degrees rather than focusing on only a single one. That is, if we decided to ignore all the *light* severity 1 incidents and start focusing on the severity 2 group, we would naturally also want to cover the higher incident groups of severity 3 and 4 as well. Hence, this would lead us ultimately to a mixed severity NKDE outcome as seen in Fig. 11(e) and Fig. 11(f). For this reason, our safe route evaluation results in terms of travelled distance, travel time and carbon emissions were very similar to the ones seen in Fig. 9, showing the same descending trend. For the case of considering only the severity 3 and/or 4 incidents, the safe route metrics were very close to the ones of the shortest routes, which can be attributed to the compact density hotspots and the overall very low density values around them.

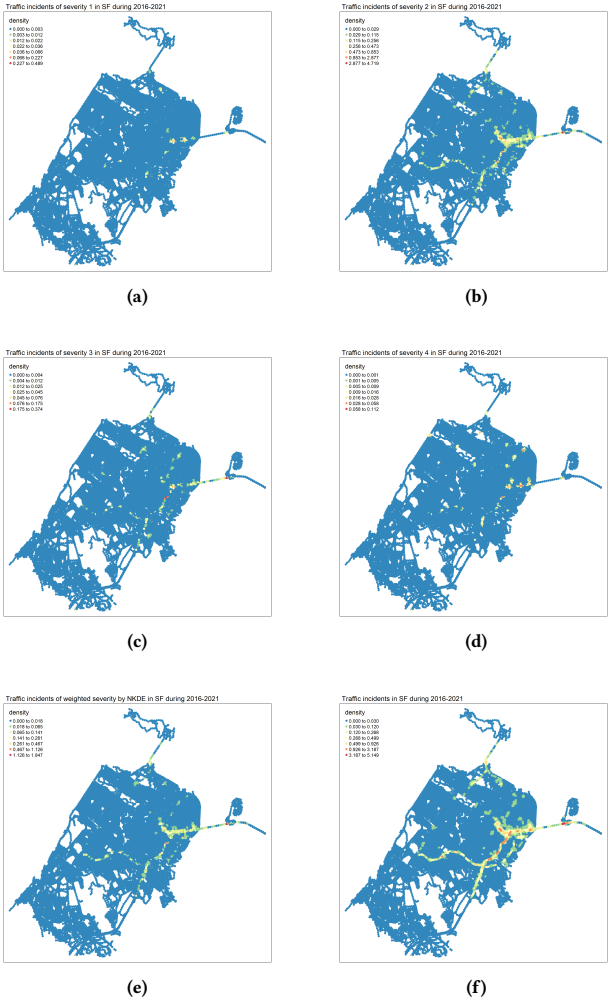


Figure 11: (a,b,c,d) NKDE based on traffic incident severity 1, 2, 3, and 4 respectively, (e) Severity-Weighted NKDE, (f) NKDE on all degrees of severity.

3.4 Temporal Network Kernel Density Estimation (TNKDE) Analysis

Traffic data are highly dynamic and typically come with a temporal dimension. Traffic accident data are no exception. There are many different aspects based on which one can observe and analyze the temporal component, such as seasonality, periodicity and frequency by month, day, weekend, time of day, to name but a few. We were particularly interested in identifying daily patterns by the time of day. Regardless the day, although the day itself can be a significant factor itself. Fig. 12 presents the hourly distribution on a San Francisco traffic accident data sample. There is a clear increasing trend of traffic accidents reaching its peak around 4pm that most likely reflects the rush hour traffic volume. A much smaller peak can be slightly identified at 7am and 9am that could probably also be attributed to the higher commute traffic. One could easily apply

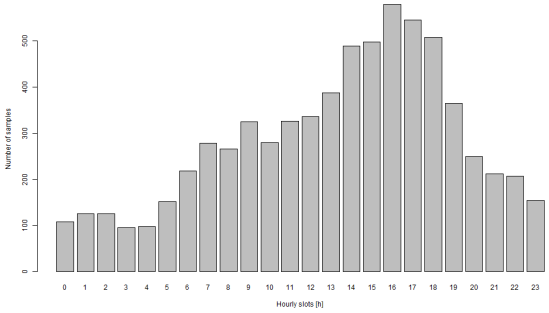


Figure 12: Temporal histogram of the San Francisco accident data sample based on hourly slots.

a 1-dimensional KDE on the temporal data and get similar results, probably a multi-modal distribution or a Gaussian Mixture. But in our case, it would be more interesting to compute a spatio-temporal KDE and in particular, a *Temporal Network KDE (TNKDE)*.

Such combined KDE functions for a location point $p(x, y)$ at time t can be approximated by multiplying the individual kernels and bandwidths as seen in Eq. 6 and which is found in [27].

$$TNKDE(p,t) = \frac{1}{bw_{Net} \cdot bw_{Time}} \cdot \sum_{i=1}^{n} (k_{Net}(d(p,l_i), bw_{Net}) \cdot k_{Time}(d(t,t_i), bw_{Time})) \quad (6)$$

where k_{Net} is the network kernel as we had in NKDE, k_{Time} the temporal kernel, bw_{Net} and bw_{Time} the corresponding bandwidths, n the number of points of interest (e.g., traffic accidents), i an incident at location l_i and time t_i, and $d(p, l_i)$ and $d(t, t_i)$ the distance in time and space between p and i. Lucky for us, the R *spNetwork* library provides us with this implementation[7]. As with NKDE before, we experimented with various kernels and bandwidths. We decided for the same Quartic kernel, a bandwidth of 300m for the network kernel and a bandwidth of about 45 minutes for the temporal kernel. Fig. 13 illustrates a sample of the resulting TNKDE snapshots, each representing a different time of day, which in this case is a $(1/70)^{th}$ snapshot of a 24-hours long day. We can see that each time slot comes mostly with its own individual small hotspots. A few of them, like Fig. 13(a), (g) and (i), seem to be more representative of the NKDE picture we've seen in the previous section. In general, while NKDE is able to identify accident-prone roads and road segments, TNKDE goes one step further by offering a more condensed and concrete view into the data and identifies specific parts of roads or intersections that stand out as particularly risky at certain times of day. Although, we haven't dived deeper into the TNKDE, it is apparent that the insights gained from such a TNKDE-based approach can help further improve and fine-tune the safe route generation process and therefore, it is part of our future work.

[7] https://jeremygelb.github.io/spNetwork/articles/TNKDE.html

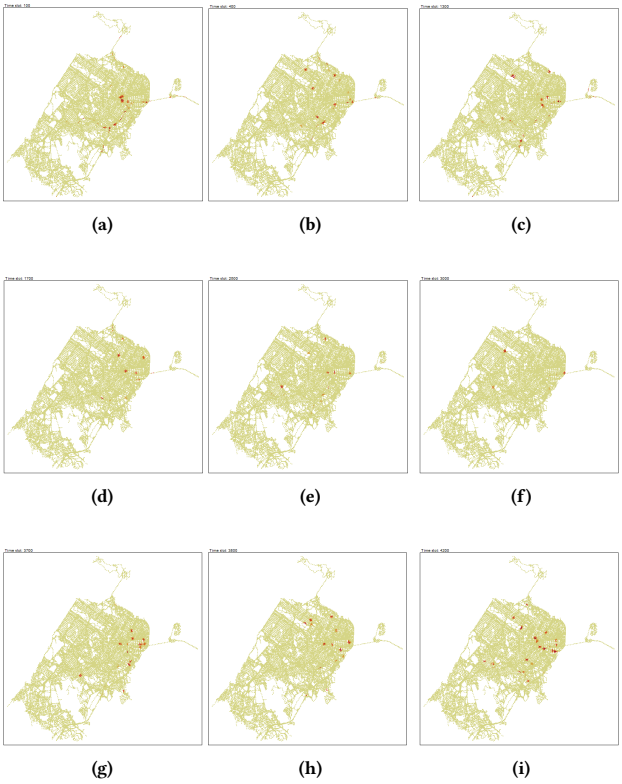


Figure 13: Example TNKDE snapshots where each image represents a different time of day and comes with its own micro-hotspots.

4 CONCLUSION AND FUTURE WORK

In this work, we explore a *Network Kernel Density Estimation (NKDE)* based approach for identifying traffic accident-prone streets on a road network and its capability of generating safer routes. We evaluate the approach using a US traffic accident dataset for the city of San Francisco in terms of travel distance, travel time and carbon emissions against the commonly in navigation systems found fastest path solution. Moreover, working our way towards the proposed final NKDE-based approach, we also go through the typical planar KDE as well as the K-Means clustering method and highlight their limitations for our scenario. Finally, this work investigates the possibility of taking the severity of the traffic incidents into account, as well as the role of time into the final results. For the latter, we make a first dive into the spatio-temporal *Temporal Network Kernel Density Estimation (TNKDE)* and discuss our findings. Both the NKDE and the TNKDE prove to be valuable when it comes to generating safer routes on a road network based on historical data.

However, the reliance on historical data defines two of the most important limitations of such data-driven methods. Namely, first, their necessity for finding the respective data that sufficiently cover the regions of interest. And second, the need for fresh, up-to-date data that reflect the most recent road and traffic flow conditions and which would help prevent potential data drift issues. Moreover, considering actual real-time signals (such as traffic signals) should further enhance the predictability of risk regions and thus lead to more accurate safer routes.

For above reasons, in our future work, we plan to further evaluate NKDE and TNKDE with additional historical and real-time factors, such like road quality flags and population density (as both have been proven to be important in past studies), as well as weather and traffic information (volume, flow, congestion flags and similar events) respectively, and how these might be combined with state of the art routing algorithms such as with customizable route planning methods and contraction hierarchies.

REFERENCES

[1] Jyoti Agarwal, Renuka Nagpal, and Rajni Sehgal. 2013. Crime analysis using k-means clustering. *International Journal of Computer Applications* 83, 4 (2013).
[2] Aya Hamdy Ali, Ayman Atia, and Mostafa-Sami M Mostafa. 2017. Recognizing driving behavior and road anomaly using smartphone sensors. *International Journal of Ambient Computing and Intelligence (IJACI)* 8, 3 (2017), 22–37.
[3] Saad Aljubayrin, Jianzhong Qi, Christian S Jensen, Rui Zhang, Zhen He, and Zeyi Wen. 2015. The safest path via safe zones. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 531–542.
[4] Yash S Asawa, Samarth R Gupta, and Nikhil J Jain. 2020. User Specific Safe Route Recommendation System. *International Journal of Engineering Research Technology (IJERT)* (2020).
[5] Murat Bayrak and Vikash V Gayah. 2021. Identification of optimal left-turn restriction locations using heuristic methods. *Transportation research record* 2675, 10 (2021), 452–467.
[6] Allan M de Souza, Torsten Braun, Leonardo C Botega, Raquel Cabral, Islene C Garcia, and Leandro A Villas. 2019. Better safe than sorry: a vehicular traffic re-routing based on traffic conditions and public safety issues. *Journal of Internet Services and Applications* 10, 1 (2019), 1–18.
[7] Haluk Eren, Semiha Makinist, Erhan Akin, and Alper Yilmaz. 2012. Estimating driving behavior by a smartphone. In *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 234–239.
[8] Kaiqun Fu, Yen-Cheng Lu, and Chang-Tien Lu. 2014. Treads: A safe route recommender using social media mining and text summarization. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 557–560.
[9] Esther Galbrun, Konstantinos Pelechrinis, and Evimaria Terzi. 2016. Urban navigation beyond shortest route: The case of safe paths. *Information Systems* 57 (2016), 160–171.
[10] Jérémy Gelb. 2021. spNetwork, a package for network kernel density estimation. *The R Journal* (2021). https://doi.org/10.32614/RJ-2021-102
[11] Thomas F Golob and Wilfred W Recker. 2001. Relationships among urban freeway accidents, traffic flow, weather and lighting conditions. (2001).
[12] Abdeltawab M Hendawi, Aqeel Rustum, Amr A Ahmadain, David Hazel, Ankur Teredesai, Dev Oliver, Mohamed Ali, and John A Stankovic. 2017. Smart personalized routing for smart cities. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 1295–1306.
[13] Abdeltawab M Hendawi, Aqeel Rustum, Amr A Ahmadain, Dev Oliver, David Hazel, Ankur Teredesai, and Mohamed Ali. 2016. Dynamic and personalized routing in prego. In *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, Vol. 1. IEEE, 357–360.
[14] Abdeltawab M Hendawi, Aqeel Rustum, Dev Oliver, David Hazel, Ankur Teredesai, and Mohamed Ali. 2015. Multi-preference time dependent routing. *Technical Report UWT-CDS-TR-2015–03-01, Center for Data Science, Institute of Technology, University of Washington, Tacoma, Washington, USA* (2015).
[15] Abdeltawab M Hendawi, Eugene Sturm, Dev Oliver, and Shashi Shekhar. 2013. CrowdPath: a framework for next generation routing services using volunteered geographic information. In *International Symposium on Spatial and Temporal Databases*. Springer, 456–461.
[16] Fariha Tabassum Islam, Tanzima Hashem, and Rifat Shahriyar. 2021. A privacy-enhanced and personalized safe route planner with crowdsourced data and computation. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 229–240.
[17] Antonios Karatzoglou. 2020. Applying topographic features for identifying speed patterns using the example of critical driving. In *Proceedings of the 13th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. 1–4.
[18] Jaewoo Kim, Meeyoung Cha, and Thomas Sandholm. 2014. Socroutes: safe routes based on tweet sentiments. In *Proceedings of the 23rd International Conference on World Wide Web*. 179–182.

[19] John Krumm and Eric Horvitz. 2017. Risk-aware planning: Methods and case study for safer driving routes. In *Twenty-Ninth IAAI Conference*.
[20] Irina Makarova, Anton Pashkevich, and Ksenia Shubenkova. 2017. Safe routes as one of the ways to reduce the number of road accidents victims. In *Scientific And Technical Conference Transport Systems Theory And Practice*. Springer, 73–84.
[21] Sobhan Moosavi, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. 2019. Accident risk prediction based on heterogeneous sparse data: New dataset and insights. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 33–42.
[22] Anne Vernez Moudon, Orion Stewart, Lin Lin, et al. 2010. *Safe routes to school (SRTS) statewide mobility assessment study: phase I report.* Technical Report. Washington State Transportation Center.
[23] Atsuyuki Okabe and Kokichi Sugihara. 2012. *Spatial analysis along networks: statistical and computational methods*. John Wiley & Sons.
[24] World Health Organization. 2019. *Global Status Report on Road Safety 2018*. World Health Organization. https://books.google.com/books?id=uHOyDwAAQBAJ
[25] Sarbaz Othman, Robert Thomson, and Gunnar Lannér. 2009. Identifying critical road geometry parameters affecting crash rate and crash type. In *Annals of advances in automotive medicine/annual scientific conference*, Vol. 53. Association for the Advancement of Automotive Medicine, 155.
[26] Sarah Otmani, Thierry Pebayle, Joceline Roge, and Alain Muzet. 2005. Effect of driving duration and partial sleep deprivation on subsequent alertness and performance of car drivers. *Physiology & behavior* 84, 5 (2005), 715–724.
[27] Benjamin Romano and Zhe Jiang. 2017. Visualizing traffic accident hotspots based on spatial-temporal network kernel density estimation. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.
[28] Romi Satria and María Castro. 2016. GIS tools for analyzing accidents and road design: a review. *Transportation research procedia* 18 (2016), 242–247.
[29] Sumit Shah, Fenye Bao, Chang-Tien Lu, and Ing-Ray Chen. 2011. Crowdsafe: crowd sourcing of crime incidents and safe routing on mobile devices. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 521–524.
[30] Bernard W Silverman. 1986. *Density estimation for statistics and data analysis*. New York: Chapman and Hall.
[31] Shivangi Soni, Venkatesh Gauri Shankar, and Sandeep Chaurasia. 2019. Route-the safe: A robust model for safest route prediction using crime and accidental data. *Int. J. Adv. Sci. Technol* 28, 16 (2019), 1415–1428.
[32] Orion T Stewart. 2018. Safe routes to school (SRTS). In *Children's Active Transportation*. Elsevier, 193–203.
[33] Kokichi Sugihara, Toshiaki Satoh, and Atsuyuki Okabe. 2010. Simple and unbiased kernel function for network analysis. In *2010 10th International Symposium on Communications and Information Technologies*. IEEE, 827–832.
[34] Boon Ean Teoh, SG Ponnambalam, and Nachiappan Subramanian. 2018. Data driven safe vehicle routing analytics: a differential evolution algorithm to reduce CO2 emissions and hazardous risks. *Annals of Operations Research* 270, 1 (2018), 515–538.
[35] William N Venables and Brian D Ripley. 2013. *Modern applied statistics with S-PLUS*. Springer Science & Business Media.
[36] Eleni I Vlahogianni and Emmanouil N Barmpounakis. 2017. Driving analytics using smartphones: Algorithms, comparisons and challenges. *Transportation Research Part C: Emerging Technologies* 79 (2017), 196–206.
[37] P Wackrill and C Wright. 2001. Identifying safe routes to school. In *UNIVERSITIES TRANSPORT STUDY GROUP 33RD ANNUAL CONFERENCE, HELD JANUARY 2001, OXFORD, UK-CONFERENCE PAPERS-VOLUME 2*.
[38] Zhixiao Xie and Jun Yan. 2008. Kernel density estimation of traffic accidents in a network space. *Computers, environment and urban systems* 32, 5 (2008), 396–406.
[39] Kazuyoshi Yajima, Kenji Ikeda, Masamitsu Oshima, and Tokio Sugi. 1976. Fatigue in automobile drivers due to long time driving. *SAE Transactions* (1976), 247–253.
[40] Guangnan Zhang, Kelvin KW Yau, Xun Zhang, and Yanyan Li. 2016. Traffic accidents involving fatigue driving and their extent of casualties. *Accident Analysis & Prevention* 87 (2016), 34–42.
[41] Min Zhou and Virginia P. Sisiopiku. 1997. Relationship Between Volume-to-Capacity Ratios and Accident Rates. *Transportation Research Record* 1581, 1 (1997), 47–52. https://doi.org/10.3141/1581-06
[42] Konstantinos G Zografos and Christian F Davis. 1989. Multi-objective programming approach for routing hazardous materials. *Journal of Transportation engineering* 115, 6 (1989), 661–673.