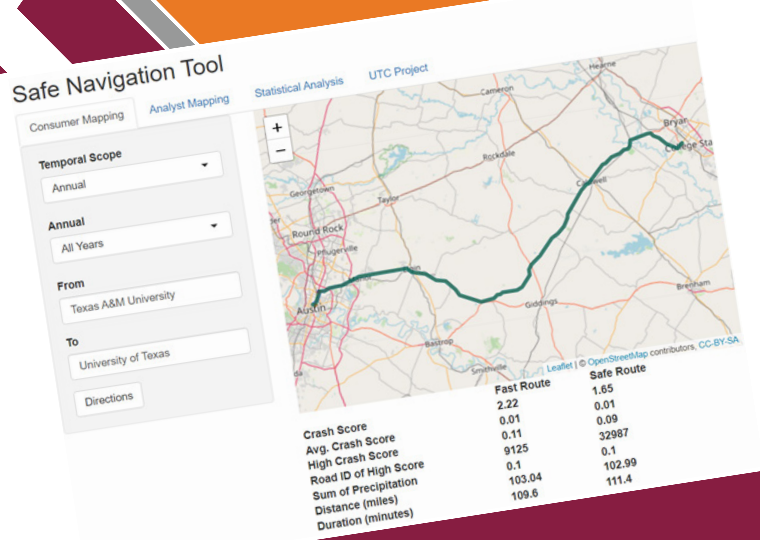
Developing AI-driven Safe Navigation Tool

August 2023 | Final Report



SAFE-D
SAFETY THROUGH DISRUPTION

VIRGINIA TECH TRANSPORTATION INSTITUTE
VIRGINIA TECH

Texas A&M Transportation Institute

San Diego State University
Leadership Starts Here

Disclaimer

TECHNICAL REPORT DOCUMENTATION PAGE

<table>
<tr><td>1. Report No.
06-002</td><td>2. Government Accession No.</td><td>3. Recipient's Catalog No.</td></tr>
<tr><td colspan="2" rowspan="2">4. Title and Subtitle
Developing Artificial Intelligence Driven Safe Navigation Tool</td><td>5. Report Date
August 2023</td></tr>
<tr><td>6. Performing Organization Code:</td></tr>
<tr><td colspan="2">7. Author(s)
Subasish Das
Ioannis Tsapakis
Yanmo Weng
Darren Torbic
Soheil Sohrabi
Xinyue Ye
Shoujia Li</td><td>8. Performing Organization Report No.
Report 06-002</td></tr>
<tr><td colspan="2" rowspan="2">9. Performing Organization Name and Address:
Safe-D National UTC
Texas A&M Transportation Institute
The Texas A&M University System
College Station, Texas 77843-3135</td><td>10. Work Unit No.</td></tr>
<tr><td>11. Contract or Grant No.
69A3551747115/06-002</td></tr>
<tr><td colspan="2" rowspan="2">12. Sponsoring Agency Name and Address
Office of the Secretary of Transportation (OST)
U.S. Department of Transportation (US DOT)</td><td>13. Type of Report and Period
Final Research Report
August 2021-August 2023</td></tr>
<tr><td>14. Sponsoring Agency Code</td></tr>
<tr><td colspan="3">15. Supplementary Notes
This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program, and, in part, with general revenue funds from the State of Texas.</td></tr>
<tr><td colspan="3">16. Abstract
Popular navigation applications such as Google Maps and Apple Maps provide distance-based or travel time-based alternative routes with no real-time risk scoring. There is a need for a real-time navigation system that can provide the data-driven decision on the safest path or route. By leveraging data from a diverse range of historical and real-time sources, this study successfully developed a user interface for a navigation tool or application that offers informed and data-driven decisions regarding the safest navigation options. The interface considers multiple scoring factors, including safety, distance, travel time, and an overall scoring metric. This study made a distinctive and valuable contribution by designing and implementing a robust safe navigation tool driven by artificial intelligence. Unlike existing navigation tools that offer multiple uninformed route options, this tool provides users with an informed decision on the safest route. By leveraging advanced AI algorithms and integrating various data sources, this navigation tool enhances the accuracy and reliability of route selection, thereby improving overall road safety and ensuring users can make informed decisions for their journeys.</td></tr>
<tr><td colspan="2">17. Key Words
Artificial Intelligence, Navigation,</td><td>18. Distribution Statement
No restrictions. This document is available to the public through the Safe-D National UTC website, as well as the following repositories: VTechWorks, The National Transportation Library, The Transportation Library, Volpe National Transportation Systems Center, Federal Highway Administration Research Library, and the National Technical Reports Library.</td></tr>
<tr><td>19. Security Classif. (of this report)
Unclassified</td><td>20. Security Classif. (of this page) Unclassified</td><td>21. No. of Pages
41   22. Price
$0</td></tr>
</table>

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Abstract

Popular navigation applications such as Google Maps and Apple Maps provide distance-based or travel time-based alternative routes with no real-time risk scoring. There is a need for a real-time navigation system that can provide a data-driven decision on the safest path or route. By leveraging data from a diverse range of historical and real-time sources, this study successfully developed a user interface for a navigation tool or application that offers informed and data-driven decisions regarding the safest navigation options. The interface considers multiple scoring factors, including safety, distance, travel time, and an overall scoring metric. This study made a distinctive and valuable contribution by designing and implementing a robust safe navigation tool driven by artificial intelligence (AI). Unlike existing navigation tools that offer multiple uninformed route options, this tool provides users with an informed decision on the safest route. By leveraging advanced AI algorithms and integrating various data sources, this navigation tool enhances the accuracy and reliability of route selection, thereby improving overall road safety and ensuring users can make informed decisions for their journeys.

Acknowledgements

This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program.

Table of Contents

List of Figures

List of Tables

Introduction

Navigation has a rich history believed to have originated with seafaring and later extended to land, aeronautics, and space navigation. In all cases, navigators determined their position relative to familiar locations or patterns, using methods such as dead reckoning or celestial navigation. As road signage limitations became apparent, navigation systems naturally gained popularity among drivers. The first car navigation system, capable of guiding drivers with the help of a rolling map, was introduced in 1910 (Jones Live Map Meter Company, 1910). Since then, car navigation systems have undergone significant advancements, including the introduction of built-in or portable devices. Not only have these devices evolved, but system capabilities and underlying algorithms have also been modified. The invention of the GPS in 1973 (Bray, 2014; Canales, 2018) had a profound impact on the car navigation industry, transitioning systems from dead reckoning to satellite-based navigation, albeit almost two decades after GPS's introduction. Another significant improvement was the inclusion of route-finding and route-planning features within in-car navigation. With the emergence of smartphones and navigation apps, advanced road navigation systems became nearly ubiquitous. Notably, travel time calculation based on traffic conditions and guiding users through the shortest path became an innovative feature. The route planning relied on shortest pathfinding algorithms (Gallo & Pallottino, 1988). Present-day road navigation systems go beyond finding the shortest path and now consider factors such as fuel efficiency and environmental friendliness in their route planning (Ding et al., 2017; Holden et al., 2020; Zeng et al., 2017).

Since the 1960s, road navigation systems have been recognized as a driving assistance technology that enhances safety (Auer et al., 2016). Initially, these systems provided basic turn-by-turn guidance to aid drivers in making safer driving decisions, such as maneuvering and changing lanes. More recent iterations of road navigation systems now also inform drivers about potential road hazards, including lane closures, changing speed limits, floods, and crashes. However, despite these advancements, route planning still does not prioritize safety. In a study conducted by Sohrabi and Lord (2022), the authors investigated the safety of suggested routes by navigation apps and found that the fastest route was not necessarily the safest. Their research revealed that a modest increase of 8% in travel time could result in a 23% reduction in the likelihood of being involved in a crash. The authors argued that road navigation primarily focuses on minimizing travel time, sometimes at the expense of directing drivers through local roads with inadequate geometry design, limited road markings and signage, increased interactions with vulnerable road users like cyclists and pedestrians, and heightened traffic conflicts, among other factors (Sohrabi & Lord, 2022).

Route guidance systems (RGSs) have been considered driving assistance technologies since 1960. RGSs were initially used in GPS devices but have had a recent transformation into mobile applications. The goals of RGSs were limited to driver assistance and travel time reduction. Online or mobile navigation tools or RGSs such as Google Maps, Apple Maps, Waze, and MapQuest provide the shortest or fastest route between origin and destination locations, with some other options such as avoiding toll roads or avoiding freeways. None of these applications provide information about safety. The fastest route is often associated with disruptions such as entering or exiting ramps in a quick fashion. On the other hand, the shortest route can be associated with other nuisances such as narrow lanes, lack of lighting, and other poor geometric design features.

There is a need for a safe navigation tool that can provide accurate measures of safety by applying prior historical data and other key associated features in artificial intelligence (AI)-driven algorithms. The AI-driven safe navigation tool will have the applicability of evaluating roadway risk scores based on historical traffic crash data with a real-time inflow of data such as weather and incident information. Based on the acquired real-time information and the backend AI-driven models, the tool will suggest the safest path from a set of origins and destinations. When the tool is used in real-time, it will have the applicability of querying relevant spatiotemporal data to produce real-time risk scoring. In addition, the tool will consider roadway and environmental features such as curvature, frequent ramp exits or entrances, frequent merging scenarios, presence of shoulders, the time of the day, lighting conditions, and weather conditions to determine the safest route. The unique contribution of a robust AI-driven safe navigation tool is to provide roadway users the ability to make an informed decision of the safest route instead of providing several uninformed decisions, as current navigation tools do.

This study aims to address the current research gap by developing an AI-driven safe navigation tool. In this study, the research team conducted a thorough literature review and assessment of available tools or applications, collected and integrated multiple datasets, including crash, roadway, weather, and traffic data, demographic data, vehicle trajectories, incidents, and crowdsourced data from various sources, performed variable selection for modeling, determined the suitable AI algorithms, and developed an application tool that provides the safest real-time routing option by predicting scores based on safety, distance, and travel time. The research team used a wide range of databases to determine the most suitable datasets.

Literature Review

Safety Consideration

The overall literature aims to optimize route safety, but specific objectives vary based on safety definitions. Safe routing differs for vehicles, non-motorists, and public transportation. Vehicle safety considers time, distance, and critical factors. Non-motorist safety also considers these along with walkability, crime scores, and additional information. The literature identified five definitions of safety: crime risk, health risk, vehicle crash risk, pedestrian or cyclist crash risk, and hazardous materials (HAZMAT) transportation risk. Around 41% of the studies analyzed vehicle crash risk, with 21% examining crashes involving pedestrians or cyclists. Crime risk studies made up 25%, while health risks and transportation risks associated with HAZMAT accounted for 7% of the literature.

Public safety literature contained information on safe route-finding. The literature also looked at locating a route (primarily in an urban setting) with minimum risk of being a victim of a crime (Utamima & Djunaidy, 2017; Byon et al., 2010; Mata et al., 2016; Kaur et al., 2021; Radojičić et al., 2018; Bura et al., 2019; Soni et al., 2019; Alpkoçak & Cetin, 2020; Levy et al., 2020; Puthige et al., 2021; Galbrun et al., 2016). The route with the minimum number of historical crimes was primarily introduced as the safest route (Alpkoçak & Cetin, 2020; Byon et al., 2010; Kaur et al., 2021; Mata et al., 2016), but some studies differentiated the type of crime and target specified crimes for their safe route-finding analysis (for example, robbery; Radojičić et al., 2018) and crimes that could result in injury or death (Galbrun et al., 2016; Levy et al., 2020; Puthige et al., 2021; Soni et al., 2019; Utamima & Djunaidy, 2017).

The literature also studied the safest route to minimize the risk of exposure to COVID-19 (Mishra et al., 2021; Cantarero et al., 2021; Khanfor et al., 2020), primarily through maintaining social distancing and avoiding crowded communities or zones. The literature review also identified several methods used to identify and avoid high-exposure hotspots. Mishra et al. (2021) identified medical zones, high-density residential areas, and roads with a higher potential for connectivity of people and COVID-19 exposure hotspots. Khanfor et al. (2020) introduced a method using internet-of-things devices in smart cities for finding pedestrian routes that help individuals avoid areas where social distancing is not properly followed. Cantarero et al. (2021) defined high-exposure areas as areas that have a high density of population and occupations.

A large portion of the literature focused on the path that reduces the likelihood of car crashes (Ito & Koji, 2020; Li et al., 2014; Sarraf & McGuire, 2018; Abdelhamid et al., 2016; El-Wakeel et al., 2018; He & Qin, 2017; Hoseinzadeh et al., 2020; Zhou et al., 2017; Abdelrahman et al., 2019; Hayes et al., 2020; Krumm & Horvitz, 2017; Li et al., 2016; Takeno et al., 2016; Kamal Alsheref, 2019; Soni et al., 2019; Liu et al., 2017; Puthige et al., 2021; Chen & Lou, 2021). Typically, the safest route is the route with lower historical crash rates (Li et al., 2016), but some studies also consider the severity of crashes. Pedestrian and cyclist safety are also studied in the literature, with pedestrian and cyclist safe routes aiming to maximize the safety of pedestrians and cyclists regarding the risk of road crashes (Santhanavanich et al., 2020; Shah et al., 2020; Bao et al., 2017; Ouyang et al., 2014; Yew et al., 2010; Kusano & Inoue, 2013; Chandra, 2014; Shubenkova et al., 2018; Lozano Domínguez & Mateo Sanguino, 2021). Additional data inputs such as sidewalk presence, the network of sidewalks, trails, and walkability scores are also typically considered for the safe routing of non-motorists. Lighting (Ouyang et al., 2014) and weather conditions (Shah et al., 2020), among others, have been associated with the risk of crashes.

Route-finding also included the HAZMAT transportation risk (Eren & Tuzkaya, 2021; Preda et al., 2013). HAZMAT transportation route-finding was developed for various types of HAZMAT by Preda et al. (2013). Eren and Tuzkaya (2021) proposed safe route-finding for COVID-19-related HAZMAT. For HAZMAT, the safest route is usually considered to be a route with less risk of HAZMAT exposure and emissions.

Safety Measurements

The reviewed studies displayed considerable diversity in terms of the measurement and quantification of safety. To summarize, safety was evaluated using different approaches including (1) data-driven analyses of crash frequency, crash rate, or incident/crash probability, (2) a scoring method, and (3) safety-indicator variables used as a substitute for risk.

Data-driven Measurement of Safety

The first group of approaches commonly utilized historical crash data for their analyses. Takeno et al. (2016) employed the dangerous point and degree of risk as safety measures to propose a safe route that avoids high-crash frequency areas. Hoseinzadeh et al. (2020) measured safety based on the average crash number, route volatility, and driving style and introduced an impedance index that combines safety and travel time for route selection. Zhou et al. (2017) defined the danger index to represent collision density on streets, indicating the level of danger in a particular street. Puthige et al. (2021) initially defined a crash score based on the severity of a crime and assigned weights to different types of crimes, resulting in a crime score that could be used to calculate a final danger index. Santhanavanich et al. (2020) incorporated historical traffic crash data into a geodatabase and identified high-risk areas or crash hot spots that their route planning algorithm aimed to avoid. Utamima and Djunaidy (2017) defined dangerous points within a specific radius based on crime data, including type, location, and description, which their route selection algorithm aimed to avoid. Yew et al. (2010) measured safety by calculating the ratio of reported incidents to exposure, treating the risk rate as a probability for the route-finding algorithm. Hayes et al. (2020) used the crash rate as a representation of safety, determined by dividing the number of crashes on a road in the past 12 months by the maximum number of crashes on a single road in the selected region. Levy et al. (2020) incorporated safety, defined by the average distance from previously known crime spots during the learning phase, into the reward function of a reinforcement learning algorithm, which generated routes that avoided the crime points. Alpkoçak and Cetin (2020) compared and measured the safety of regions using the crime rate, which represents the ratio of incidents to the population in a given region. Bura et al. (2019) measured safety by considering the number of crime records with the aim of finding routes with fewer crime records.

Advanced data-driven models were utilized to estimate the likelihood of an incident through prediction models. Mata et al. (2016) used the Bayes model to categorize crimes and calculate the probability of events occurring at specific times, days, and locations. In their study, Radojičić et al. (2018) defined risk as the probability of an event happening on a specific route, which varied as the vehicle traveled further. Abdelhamid et al. (2016) examined the likelihood of being involved in crashes by analyzing the characteristics of the road segment and the behavior of the vehicle driver. Similarly, Abdelrahman et al. (2019) determined the probability of crashes by considering the number of crashes, near-crashes, and baseline events observed in the driving environment. Krumm and Horvitz (2017) used independent Bernoulli trials to estimate the hourly probability of a single-vehicle crash on a specific road segment. Galbrun et al. (2016)

assigned risk scores to edges based on the probability of crime events occurring on the corresponding road segment.

Scoring Technique

The second group of approaches employed a scoring technique to measure safety, utilizing a scaled value. Kaur et al. (2021) assigned safety scores ranging from 0 to 15 to road segments. They adjusted these scores based on specific parameters, such as the presence of a police station or passing through deserted regions. Higher scores indicated safer road segments, thereby determining the safest route. Bao et al. (2017) assessed individual safety factors and assigned safety scores to segments, where higher scores indicated safer routes. The overall route safety score was determined by summing up sub-route scores, which were then used as weights in the shortest route-finding algorithm for optimal route determination. Mishra et al. (2021) assessed safety using various hazard factors, assigning numerical values to each. Byon et al. (2010) focused on crime risk, tallying the number of deaths associated with each zone. The zone with the highest death count received a rating of 5 and the lowest a rating of 1. Soni et al. (2019) used an overall numeric risk score to measure safety. By considering all points and regions, they estimated the average crime and crash scores. The risk score was obtained by summing up these averaged scores. Shah et al. (2020) studied pedestrian safety by using traffic and weather data to assess safety. They defined a safety factor based on traffic and weather conditions. Eren and Tuzkaya (2021) developed a safety score for Medical Waste Management that ranged from 1 to 10. Kamal Alsheref (2019) used traffic crashes as the safety parameter and measured safety importance using Saaty's 9-point scale (Zanakis et al., 1998). The obtained safety intensity was incorporated into the weight calculation in the route selection algorithm.

Safety Indicator

The final group of approaches assessed safety using various safety indicators. Khanfor et al. (2020) assessed safety using weighted edges in a road map graph to find the safest route. Cantarero et al. (2021) developed a method to measure safety by considering citizen density, occupation level, behavior, vulnerability, and transmission risk. Shubenkova et al. (2018) assessed safety using objective and subjective parameters. Each parameter was given a numerical value and weight, and safety was calculated by multiplying these values and weights. Ouyang et al. (2014) assessed cyclist safety using factors like slope, road type, width, signs, and lighting. Each factor had different levels, and safety weights were assigned from 0 to 1 (with 1 being the safest). The safety coefficient was calculated by multiplying these weights. El-Wakeel et al. (2018) identified and categorized road surface types and anomalies, qualitatively labeling average road quality as an indicator of safety for route selection. Ito and Koji (2020) considered rain events on roads as risky situations, defining numerical traffic risk levels from 1 to 10 based on historical rain events. He and Qin (2017) used the ratio of the deceleration rate to the maximum available deceleration rate as a proxy for traffic safety, creating a safety hazard index for roadway segments and intersections. Chandra (2014) developed crash indicators for cyclists and older drivers, considering traffic attributes like speed and density, driver attributes such as perception-reaction time, and street attributes such as length and tire-to-road friction coefficient.

Route-finding Algorithms

In the literature, the Dijkstra algorithm and its variants were mainly used to find optimal routes (Santhanavanich et al., 2020; Mishra et al., 2021; Utamima & Djunaidy, 2017; Cantarero et al., 2021; Sarraf & McGuire, 2018; Abdelhamid et al., 2016; Byon et al., 2010; Zhou et al., 2017; Mata et al., 2016; Krumm & Horvitz, 2017; Takeno et al., 2016; Yew et al., 2010; Liu et al., 2017; Galbrun et al., 2016). Originally, the Dijkstra algorithm aimed to find the shortest route. Now, it also considers safety by minimizing a cost function. This modified version calculates the lowest-cost path from one starting point to all other points in a weighted, directed graph. Inputs include safety parameters and geographic data. By incorporating factors like crime rate, road slope, scenic view, and ground surface elevation, the algorithm suggests multiple safe route options. Byon et al. (2010) improved the algorithm's safety calculations, resulting in routes with lower crime rates and better road conditions. The A* algorithm was also used in the literature with a modified heuristic function (Hayes et al., 2020; Alpkoçak & Cetin, 2020). The input parameters contain the geographic information for the origin and destination points and the safety-related parameters, similar to the Dijkstra algorithm.

Ranking the shortest path and its alternatives based on safety is another method for finding the safest route (Ito and Koji, 2020; Hoseinzadeh et al., 2020; Bura et al., 2019; Bao et al., 2017; Liu et al., 2017; Puthige et al., 2021; Chandra, 2014). Machine learning (ML) can generate safety measurements and predict safe routes. Researchers have used deep reinforcement learning to find efficient routes based on factors like crime incidents, vehicle speed, and road conditions (Levy et al., 2020). The algorithm considers starting and ending coordinates and outputs the safest path.

The algorithms for safe route-finding can be classified based on their predictive/reactive nature, static/dynamic properties, and centralized/decentralized designs. The following sections provide an overview of the literature, focusing on the algorithm's specific characteristics.

Predictive vs. Reactive Algorithms

Route-finding algorithms can be categorized as reactive or predictive (Schmitt & Jula, 2006). The literature consists of 78.05% predictive algorithms and 21.95% reactive algorithms. Reactive algorithms rely on observed data (Schmitt & Jula, 2006), while predictive algorithms use models to anticipate future conditions. In crime event estimation, various methods have been employed: Gaussian kernel density estimation (Galbrun et al., 2016), Bernoulli trials for crash probability (Krumm & Horvitz, 2017), Bayes algorithm for crime probability prediction (Mata et al., 2016), and considering behavior and context for risk prediction (Abdelrahman et al., 2019). Machine learning models, such as deep reinforcement learning, can also be used, as shown by Levy et al. (2020). Predictive algorithms can incorporate safety parameters to generate an overall safety index (Li et al., 2014; Li et al., 2016; Radojičić et al., 2018).

Static vs. Dynamic Algorithms

The route-finding algorithm can be divided into two types, static and dynamic, depending on if the route-finding system reacts to real-time information (Schmitt & Jula, 2006; Herbert & Mili, 2008; Dong, 2011; Khanjary & Hashemi, 2012). Static algorithms make up 63.41% of the reviewed studies (Mishra et al., 2021; Utamima & Djunaidy, 2017; Sarraf & McGuire, 2018; Preda et al., 2013; He & Qin, 2017; Byon et al., 2010; Zhou et al., 2017; Mata et al., 2016; Radojičić et al., 2018; Hayes et al., 2020; Bura et al., 2019; Krumm & Horvitz, 2017; Takeno et al., 2016; Kamal Alsheref, 2019; Soni et al., 2019; Bao et al., 2017; Eren & Tuzkaya, 2021; Alpkoçak & Cetin, 2020; Yew et al., 2010; Levy et al., 2020; Puthige et al., 2021; Chandra, 2014; Chen & Lou, 2021; Khanfor et al., 2020; Shubenkova et al., 2018; Galbrun et al., 2016; Lozano Domínguez & Mateo Sanguino, 2021). Dynamic algorithms make up 36.59% (Santhanavanich et al., 2020; Ito & Koji, 2020; Li et al., 2014; Cantarero et al., 2021; Shah et al., 2020; Abdelhamid et al., 2016; El-Wakeel et al., 2018; Hoseinzadeh et al., 2020; Abdelrahman et al., 2019; Kaur et al., 2021; Li et al., 2016; Ouyang et al., 2014; Liu et al., 2017; Kusano & Inoue, 2013). There are both static and dynamic algorithms from each year except for 2010. The challenges in handling real-time data may account for the lower publication rate of dynamic algorithms in contrast to static algorithms.

The datasets employed in the static algorithms and dynamic algorithms can be classified into geographic information datasets, historic datasets, and real-time datasets. Geographic datasets, containing start and endpoint information along with road network data, are used for both static and dynamic route-finding. OpenStreetMap, Bing Maps, and Google Maps are example sources for these datasets (Santhanavanich et al., 2020; Mishra et al., 2021; Utamima & Djunaidy, 2017; Cantarero et al., 2021; Shah et al., 2020; Sarraf & McGuire, 2018; El-Wakeel et al., 2018; He & Qin, 2017; Zhou et al., 2017; Hayes et al., 2020; Bura et al., 2019; Krumm & Horvitz, 2017; Takeno et al., 2016; Soni et al., 2019; Alpkoçak & Cetin, 2020; Liu et al., 2017; Levy et al., 2020; Puthige et al., 2021; Khanfor et al., 2020; Galbrun et al., 2016; Lozano Domínguez & Mateo Sanguino, 2021). Static algorithms have also used historical data for safety measurements; as an example, the static algorithm-related studies in the crime risk category used historical crime records while the dynamic algorithms incorporated real-time data. The literature also used various real-time datasets, including news websites (Mata et al., 2016; Kaur et al., 2021), official reports (Li et al., 2014; Hoseinzadeh et al., 2020; Li et al., 2016), various sensors and GPS data (Ito & Koji, 2020; El-Wakeel et al., 2018; Abdelrahman et al., 2019; Ouyang et al., 2014; Kusano & Inoue, 2013), and data from application programming interfaces (APIs) (Santhanavanich et al., 2020; Cantarero et al., 2021; Shah et al., 2020; Abdelhamid et al., 2016; Liu et al., 2017).

Centralized vs. Decentralized Algorithms

Route-finding algorithms can be decentralized (Schmitt & Jula, 2006; Khanjary & Hashemi, 2012), where individual users make decisions to maximize their benefit, or centralized (Schmitt & Jula, 2006; Khanjary & Hashemi, 2012), where the aim is to optimize the benefit of all users or society. In the safe route-finding literature, decentralized algorithms dominate, as all reviewed papers use them.

Safe Route-finding in Road Navigation

Safe route-finding algorithms guide users by providing safety information on different routes. Relying solely on historical data to estimate crash risk has limitations because crashes are rare and involve factors that can change over time (Mannering, 2018). Various factors, such as road conditions, weather, user behavior, and traffic, influence crash risk (Petridou & Moustaki, 2000; Lord et al., 2021; Qiu & Nixon, 2008). Therefore, safe route-finding should consider these changing factors and offer future-oriented insights for initial route selection and alternative paths. Ideally, a dynamic, predictive algorithm is needed to provide reliable real-time crash risk predictions.

Transitioning to predictive algorithms requires crash prediction models. Traditional models assume crashes follow a binary trial and use the Poisson process for multiple vehicles (Lord et al., 2005). However, crashes typically follow a negative binomial distribution over a specific time period. These models consider environmental, behavioral, and traffic factors to estimate expected crashes (Washington et al., 2020), but they have limitations for dynamic routing. Real-time models can update and consider traffic volatility (Hossain et al., 2019). Some models predict vehicle-level crash risks (Basso et al., 2021), incorporating user-specific factors like driving experience and vehicle characteristics. These personalized models enhance risk estimations for personalized route selection.

In the literature on roadway safety, crashes serve as the primary indicator of safety (Lord et al., 2021). However, given the difficulties in accessing comprehensive and reliable crash databases, alternative measures of safety, such as time to collision and traffic conflicts, have been employed as substitutes (Lord et al., 2021). Nevertheless, scoring-based safety assessments are susceptible to subject bias. Although a range of safety metrics can be employed to evaluate the safety of individual road segments, assessing the overall safety of a route (which typically comprises multiple road segments) is essential for identifying a secure path. A comprehensive database is essential for safe route finding. Two main types are needed: (1) a road network inventory database, and (2) a database with operational measures like speed, travel time, and historical crash data for each road segment.

Table 1 discusses different data elements and sources. For additional data sources, refer to Tarko et al. (2021). Additional details of the literature review are provided in Appendix A.

Table 1. Example Data Sources for Safe Route-finding

Data Elements	Data Source
SPEED MEASURES	
Posted speed limit or travel time	State DOT, HERE, SHRP 2-RID
Avg. operating speed, percentile speed, & speed variance	State DOT, INRIX, INRIX XD, HERE, NPMRDS
Continuous 5-minute, 15-minute, hourly, daily, monthly & annual speed	State DOT, INRIX, INRIX XD, HERE, NPMRDS
Vehicle trajectory data or waypoint data	INRIX, Wejo, StreetLight
Percent of vehicles exceeding speed limit	State DOT
ROADWAY INVENTORY DATA	
Segment length	State DOT, HPMS, SHRP 2-RID, GoogleEarth
Number of lanes	State DOT, HPMS, SHRP 2-RID, GoogleEarth
Shoulder and lane width	State DOT, HPMS, SHRP 2-RID, GoogleEarth
Horizontal and vertical alignment	State DOT, HPMS, SHRP 2-RID, GoogleEarth
Median barrier	State DOT, HPMS, SHRP 2-RID, GoogleEarth
Roadside fixed objects (barrier, guardrail, poles)	State DOT, HPMS, SHRP 2-RID, GoogleEarth
Traffic control devices, pavement condition	State DOT, HPMS, SHRP 2-RID, GoogleEarth
WEATHER CHARACTERISTICS	
Continuous hourly, daily, monthly and annual precipitation & visibility data	NOAA, Road Weather Information System (RWIS)
TRAFFIC VOLUME MEASURES	
Annual average daily traffic	State DOT, TMAS, HPMS, SHRP2-RID, StreetLight Data Inc.
Hourly traffic volume	TMAS, StreetLight
CRASH MEASURES	
Crash time and date	State DOT, HSIS, SHRP 2-RID
Crash location	State DOT, HSIS, SHRP 2-RID
Crash type and severity	State DOT, HSIS, SHRP 2-RID
Crash contributing factors (e.g., speeding)	State DOT, HSIS, SHRP 2-RID
Lighting and weather conditions	State DOT, HSIS, SHRP 2-RID

Data Elements	Data Source
REAL-TIME TRAFFIC DATA FOR THE U.S.	
1.70 million cases (2016-2019) [every 90 seconds]	MapQuest Traffic Application Programming Interface (API)
0.54 million cases (2016-2019) [every 90 seconds]	Bing Map Traffic API
REAL-TIME INCIDENT DATA	
Crowdsourced data	Waze

Methodology

Data Needs for Safe Navigation Tool

Roadway inventory data typically includes information about various roadway features, such as segment length, roadbed width, median type, shoulder type, and shoulder width. In many cases, these roadway features are provided in separate layers. However, some state databases may not offer additional geometric data such as super elevation, curve radius, and posted speed limit. To obtain supplementary geometric data, researchers can utilize resources like OpenStreetMap and private data vendors like HERE.

Regarding operational measures, researchers need to gather data on two main metrics: features related to travel time and traffic volume, and features related to recurrent and non-recurrent events, such as road incidents. Figure 1 provides an overview of the databases required for static and dynamic safe route finding. Several potential open-source datasets can be obtained from entities like Departments of Transportation (DOTs), the Highway Performance Monitoring System (HPMS), the Highway Safety Information System (HSIS), the Roadway Inventory Database (RID), and the Naturalistic Driving Study (NDS) data (from the Second Strategic Highway Research Program or SHRP 2). Additionally, data from resources like Traffic Monitoring Analysis Systems (TMAS) and the National Performance Management Research Data Set (NPMRDS) can be accessed. Commercial private data vendors offer the option to purchase or acquire data. Some private data vendors, like Wejo, also provide data obtained from vehicle on-board devices, encompassing elements like trajectory, wiper usage, acceleration, deceleration, hard braking, sudden stoppage, and near collision. Real-time traffic data can be acquired from sources such as MapQuest ("Official MapQuest – Maps, Driving Directions, Live Traffic," n.d.) and Bing Map Traffic AI ("Bing Maps Dev Center – Bing Maps Dev Center," n.d.). Crowdsourced databases like Waze can serve as potential sources for real-time incidents. To design a safe routing algorithm, it is necessary to gather data from both public and private data sources. As previously noted,

Table 1, above, outlines some of the key data elements and the corresponding data sources.

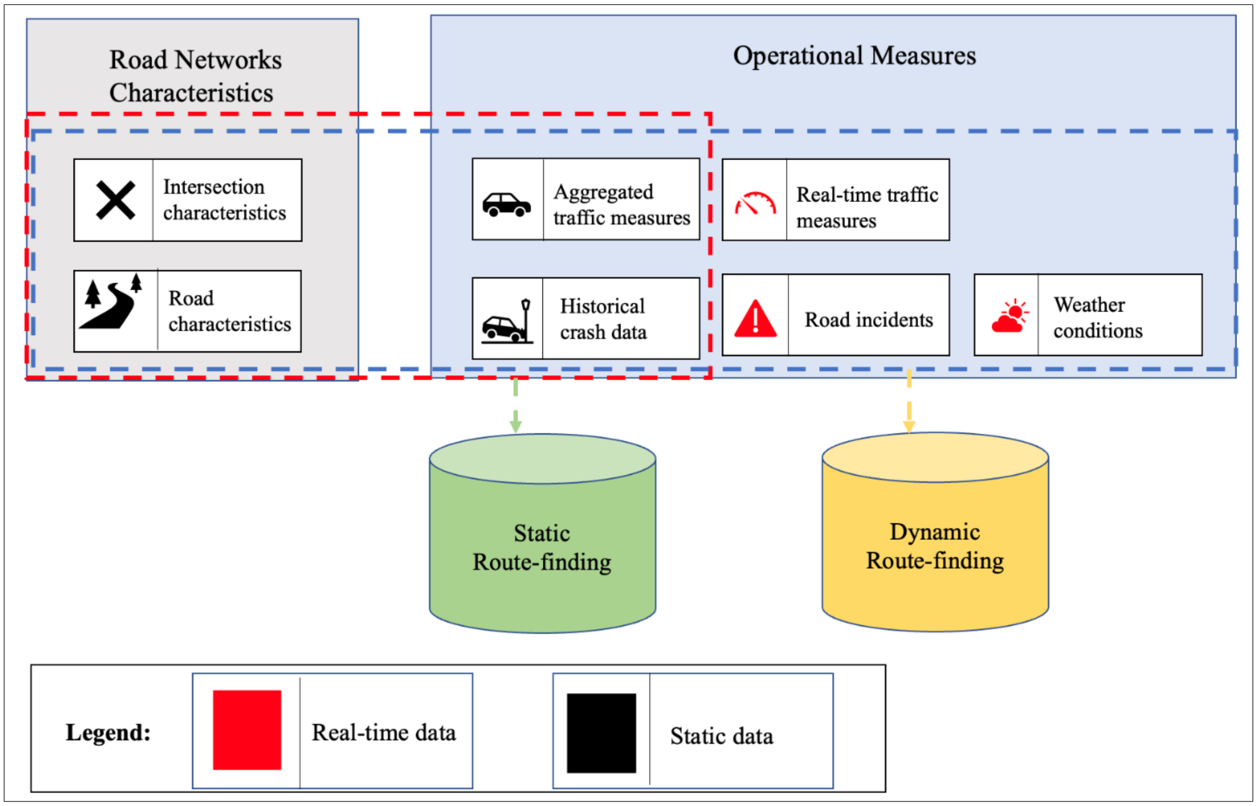


Figure 1. Static and dynamic safe route-finding data requirements.

Databases for Texas-based Case Study

The research team used Texas as a case study to develop a Texas-based AI-driven safe navigation tool. The following datasets were used for the tool. Additional details of the data preparation method are provided in Appendix B.

CRIS

The research team collected crash data from TxDOT for 5 years (2017–2021), as shown in Table 2. There are 172 fields in the Crash Records Information System (CRIS) dataset. For example, in the 2021 CRIS data, among the 552,125 unique crashes, 542,445 crashes (~98.25%) contain absolute location data in the form of latitude and longitude coordinates. The "Latitude" and "Longitude" fields are recorded to the fifth decimal place, which signifies accuracy up to 3.6 feet. For the initial spatial analysis, only the "CRASH_ID," "Latitude," and "Longitude" fields were used, as shown in Table 3.

Table 2. Overview of CRIS

	2017	2018	2019	2020	2021
Total Entries	538,739	626,514	560,835	475,132	552,125
Entries with Lat-long	499,484	531,848	543,039	465,938	542,445
Entries Missing Lat-Long	39,225	94,666	17,796	9,194	9,680
Total Number of Variables	170	170	170	173	173

Table 3. Key Attributes for Spatial Analysis

No.	Attribute Name	CRIS Field	Description
1	Crash ID	Crash_ID	System-generated unique identifying number for a crash.
2	Latitude	Latitude	Latitude map coordinate of the crash.
3	Longitude	Longitude	Longitude map coordinate of the crash.

RHiNO

The Roadway Inventory Annual Data published by the TxDOT is publicly available for download as a zip file. This database is known as Roadway Highway Inventory Network Offload (RHiNO). The zip file contains two shapefiles: one with assets and another with no assets. The primary key, all unique values, and the column of the shapefile without assets comprise the "GID" column. The shapefile with assets does not have a field with all unique values. In 2021, there were 514,480 road segments recorded in the shapefile without assets, as shown in Table 4. In the same year, there were 883,837 road segments recorded in the shapefile with assets. There were 513,885 unique values recorded in the shapefile with assets "GID" column. There were seven fields in the shapefile without assets (as shown in Table 4). There were 202 fields in the shapefile with assets.

Table 4. Overview of RHiNO Without Assets

Description	Count/Information
Total Records	514,480
Total Fields	7
Primary Key	GID
Entity	TxDOT
Source URL	https://www.txdot.gov/data-maps/roadway-inventory.html
Classification	Public Data
Datum	North American Datum 1983

NLDAS

The goal of the North American Land Data Assimilation System (NLDAS) is to construct quality-controlled and spatially and temporally consistent land-surface model (LSM) datasets from the best available observations and reanalyze the data to support modeling activities. Specifically, this system is intended to reduce errors in the stores of soil moisture and energy which are often present in numerical weather prediction models and which degrade the accuracy of forecasts. NLDAS is currently running operationally in near real-time (~4-day lag) on a 1/8th-degree grid with an hourly timestep over central North America (25–53 North; see Table 5). Retrospective hourly/monthly NLDAS datasets extend back to January 1979. NLDAS constructed a forcing dataset from a daily gauge-based precipitation analysis (temporally disaggregated to hourly using Stage II radar data), bias-corrected shortwave radiation, and surface meteorology reanalysis to drive four different LSMs to produce outputs such as surface fluxes, soil moisture, snow cover, and streamflow.

Table 5. Overview of NLDAS

Description	Information
Spatial	0.125 ° x 0.125 °
Temporal	1 hour
Range	-108,25, -92,37
crs	+proj=longlat +R=6371200 +no_defs
Time	01/01/2017 - 12/31/2021
Source URL	https://disc.gsfc.nasa.gov/datasets?keywords=NLDAS&page=1

Routing Algorithms

Popular navigation applications such as Google Maps and Apple Maps provide distance-based or travel time-based alternative routes with no real-time risk scoring. There is a need for a real-time navigation system that can supply more opportunities to avoid crash risks. This study used a Google Directions API and an Open Route Services (ORS) API to explore the route-finding algorithms with the inclusion of safety risk scores of different routes. The tool developed from this study will provide appropriate and proactive interventions that will reduce crash risk.

Google Directions API Shortest Path Algorithms

Google Maps essentially uses two graph algorithms, Dijkstra's algorithm and an A* algorithm (Mehta et al., 2019), to calculate the shortest distance from the source to the destination. Developers can use Google Directions API to use the function to display the shortest route in Google Maps. Google Directions API provides three alternative routes for users to choose the most appropriate one according to the specific situation.

Dijkstra's Algorithm

Dijkstra's algorithm is the basic algorithm in both the Google Directions API and ORS API. It is an approach to solving the single-source shortest path problem for an assignment graph (M. Noto & H. Sato, 2000). This algorithm takes out the least distant of the unvisited nodes each time and updates the distances of the other nodes with that node (Deng et al., 2012). The point of the algorithm is to repeatedly select the shortest segment every time until all of the nodes can be found.

A* Algorithm

The A* algorithm is one of the best-known path planning algorithms and can be applied to a metric or topological configuration space. It is defined as a best-first algorithm because each cell in the configuration space is evaluated by heuristic distance (Manhattan distance) of the cell to the goal node and the length of the path from the initial node to the goal node through the selected sequence of cells (Duchoň et al., 2014). Manhattan distance is the number of steps from the current node to the goal node, disregarding diagonal nodes.

ORS API Shortest Path Algorithms

The first principle to understand about the algorithm used for calculating the shortest route in the ORS API is that the algorithm is intended to minimize *travel time* rather than *travel distance*. Second, the mode of transportation selected by the user (i.e., Driving, Walking, Bicycling, etc.) and additional filters (i.e., avoid_features) will affect which algorithm is used to calculate the route. There are two algorithms used to calculate the shortest *travel time*, Contraction Hierarchies (CH) and Core-ALT (CALT).

CH Algorithm

When driving by car and not using any filters, the ORS API uses the CH algorithm to calculate the shortest *travel time* route. Using this algorithm, roads are categorized ordinally by level with international highways being considered the highest level and residential roads being considered the lowest level. At the furthest distance, the route only considers the highest level of roads for travel. As the route nears its destination, lower-level roads are considered. The Dijkstra algorithm is then used to search through the hierarchy filtered networks to determine the shortest *travel time* route to the destination.

CALT Algorithm

When implementing filters, the ORS API uses the CALT algorithm to calculate the shortest *travel time* route. When it is desirable to avoid obstructions such as rush-hour traffic jams, car accidents, or roads that are a higher safety risk to drive on, the CALT algorithm is used. For the CALT algorithm, as in other *travel time* route algorithms, each road segment has a weight assigned to it based on the avoidance of undesirable segments. The CALT algorithm incorporates dynamic weights on each segment, which consider any obstructions and adjust the route based on the avoidance of undesirable segments when an alternate route is available.

AI-Driven Safe Navigation Tool Development

AI-based Risk Scoring

In this study, the research team employed a comprehensive approach by utilizing four distinct AI models: random forest, gradient boosting, support vector regression, and CatBoost (see Appendix C for the details of the algorithms). The objective was to develop an effective risk scoring system that considers various factors such as crash data, geometric properties of roadways, and weather information. By leveraging these AI models, this study aimed to capture the complex relationships and patterns present within the dataset, ultimately generating accurate risk scores for different road segments. Each model underwent rigorous training and validation processes to ensure robust and reliable results. Upon evaluating the performance of these models, it was observed that the CatBoost algorithm outperformed the other three algorithms in terms of predictive accuracy and overall performance. As a result, the final predictive values utilized in this study were derived from the CatBoost model.

Safe Navigation Tool Development

The development tool can be accessed at: http://70.112.144.115:3838/. The interface of the tool, which used the ORS API, is shown in Figure 2,

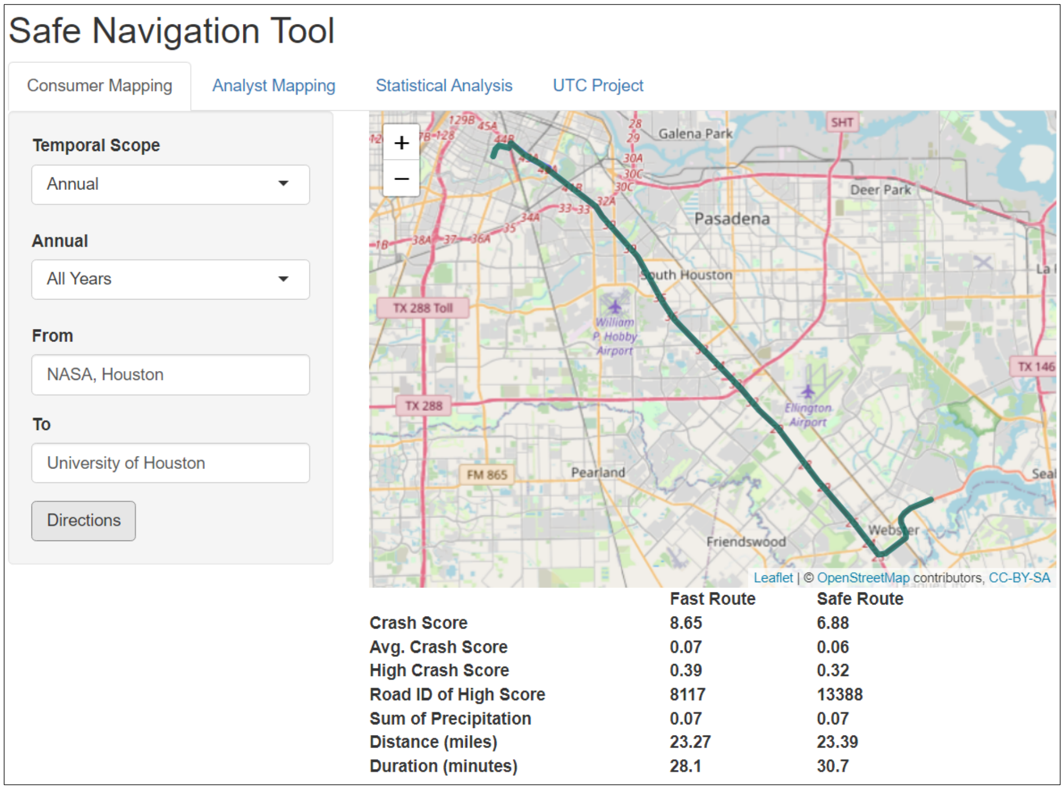


Figure 2. User interface of safe navigation tool.

The tool has four drop-down panels:

- Temporal Scope (Annual, Day of the Week, Day of the Month)
- Annual (All Years, 2017, 2018, 2019, 2020, 2021)
- From (Start location)
- To (end location)

The user interface of the developed system incorporates drop-down panels, allowing users to select their preferred temporal scope and prioritize annual durations for risk assessment. To specify the desired time period, users can input “from-to” values similar to other popular mapping applications. Upon entering this information, users are required to click the “Direction” button. Upon completion of this action, the system generates a map displaying the safest route on the right-hand side of the user interface. Complementing the map, a tabulated comparison is provided below, highlighting the distinctions between the fastest route and the safest route. This comparison encompasses several key metrics, including crash scores, average crash scores, high crash scores, road identification associated with the highest crash score, sum of precipitation, distance in miles, and duration in minutes. This user-friendly system design provides users with a comprehensive overview of the trade-offs between different routes based on safety considerations. By presenting both the visual representation of the safest route and a detailed tabulation of relevant metrics, users can make informed decisions that balance their priorities regarding travel time and road safety.

Tool Development Steps

The following steps were done to develop the tool (see Figure 3):

- The user interface for this application was implemented using the R Shiny library. In the “sidebarPanel” function, user inputs are defined using “textInput” functions, which include a variable name (e.g., “fromAddress”), a display name (e.g., “From”), and a default input value (e.g., “Texas State University"). These input fields allow users to specify their desired starting location. In the “mainPanel” function, the application displays three outputs to users. The first output is a map representation that shows the routes based on the user’s input. The map visually depicts the various routes available.
- The application's behavior is determined by the server function, which relies on user-interface data. The “fromAddress” and “toAddress” variables provided by the user are geocoded using the ORS Geocoding API to obtain precise coordinates for the specified locations. These coordinates are then utilized to retrieve the fastest route directions from the ORS Directions API. The resulting route is subsequently processed through the “safeRoute” function, which extracts relevant information and descriptive text. This data is stored as “text” and subsequently returned to the user as part of the application's response.
- The route is first converted to a “line string” shapefile. Then, an index of segments that intersect the route is created using the “st_intersects” function. Next, the index is iteratively looped to determine the combined safety score and to identify the most dangerous segment along the faster *time travel* route. Lastly, statistical data is returned from the function for further processing.

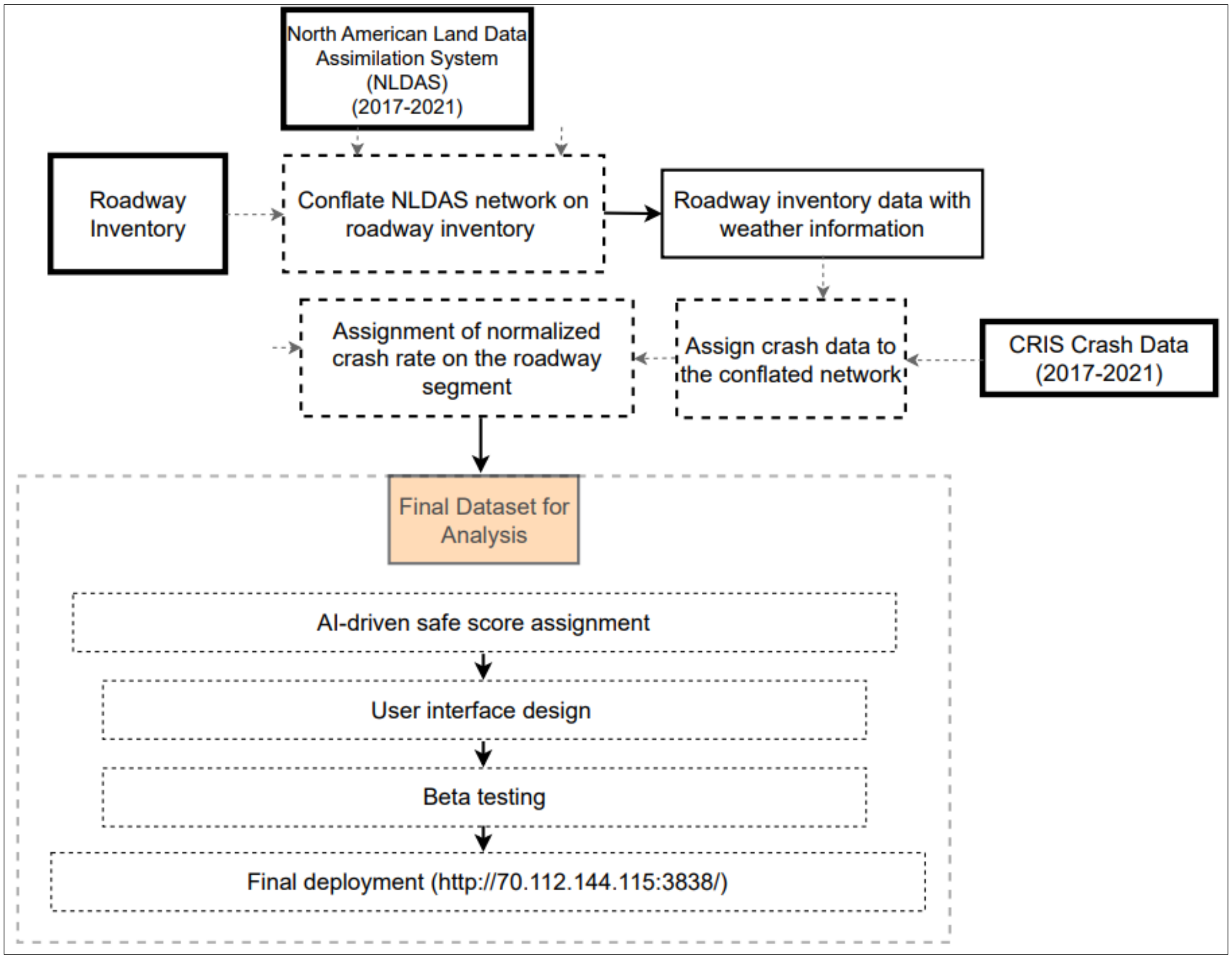


Figure 3. Framework of safe navigation tool development steps.

- The process begins by converting the route into a "line string" shapefile. Using the "st_intersects" function, an index is generated, consisting of segments that intersect with the route. Through an iterative loop, the index is analyzed to calculate the aggregated safety score and identify the segment with the highest level of danger along the faster time travel route. Finally, statistical data is extracted from the function, allowing for subsequent analysis and processing.
- The research team implemented various components into the dashboard, including the header, sidebar, and main body, with the added functionality of hiding the header and sidebar if desired. Within the main body, two "textInput" functions were integrated, allowing users to input their desired source and destination. Additionally, an "actionButton" function was included to provide users with a clickable button that triggers the display of the map. The central component, the "google_mapOutput" function, plays a pivotal role in showcasing Google Maps, complete with the identified routes and associated crash attributes.
- The server function plays a crucial role in the conversion of user interface inputs into a data frame of routes, which can then be processed and displayed on the map. Within this function, the variable "df_route" is created, containing important route details such as the source and

destination addresses. The “output$myMap” function serves as the central component responsible for determining the map’s visual display. It begins with an “if” statement to check for empty inputs, terminating the function if the condition is met. Additionally, an API key, necessary for accessing Google Maps functions and data, is included. Finally, the “google_directions” function from the “googleway” package is utilized to geocode the inputted source and destination addresses, generating encoded lines representing the route information to be displayed on the map.

- To select the safest route among the alternatives provided by the Google Directions API, a series of data processing steps were performed. Firstly, the encoded routes from Google Maps were decoded to enable further coding-based data processing. The decoding process yielded waypoints, which were then combined using the “sf_linestring” function to create polylines. Subsequently, crash data were integrated into the database, assigning the respective number of crashes to each route. The “sf_nearest_feature” function was employed to identify the line segments closest to each route, allowing for the assignment of the corresponding crash numbers. Finally, a “for” loop was implemented to identify the safest route with the minimum number of crashes. These steps facilitated the selection of the route with the optimal safety outcome.

The analysis of risk in road segments faces limitations due to the large variability in segment lengths. Although the average risk appears lower when considering the entire length of a 10,000-foot road segment, it is crucial to acknowledge that there might be localized clusters of high-risk areas. These clusters could exhibit more than 50 crashes within 100-foot sections along the 10,000-foot road segment. To address this limitation and reduce uncertainty, future research could consider dividing longer segments into shorter ones before associating crashes with their nearest segment. Additionally, while ORS available through the public API impose a distance restriction of approximately 93 miles for route calculations, this constraint can be overcome by locally hosting a private API server.

Conclusions

For real-time routing, it is essential to have up-to-date information regarding traffic conditions, road incidents (such as closures caused by flooding, crashes, and maintenance), and weather conditions. The availability of this data is crucial for the development of dynamic and predictive algorithms aimed at finding safe routes. However, this real-time information, particularly regarding traffic conditions and road closures, is not easily accessible to the public. This poses significant challenges in the development of effective route-finding algorithms. While the cost associated with collecting real-time data on traffic and road conditions can be high, crowdsourced data presents a potential alternative data source for obtaining real-time information (Amin-Naseri et al., 2018; Lin & Li, 2020). The critical balance between time and safety is crucial. When the quickest route does not align with the safest route, individuals encounter a predicament regarding whether to select the safest path or the fastest one. While the decision to prioritize either safety or time is subjective and dependent on the driver's preferences, it is anticipated to have an impact on other individuals using the road. Gaining insights into users' decision-making process when confronted with the choice between safety and travel time would facilitate proactive initiatives, such as educational activities, aimed at fostering road safety (de Leur & Sayed, 2003).

Existing navigation applications like Google Maps and Apple Maps lack real-time risk scoring, resulting in limited safety considerations when suggesting alternative routes based on distance or travel time. To address this gap, this study successfully developed a user interface for a navigation tool that leverages diverse historical and real-time data sources to offer informed and data-driven decisions regarding the safest navigation options. By considering multiple scoring factors such as safety, distance, travel time, and an overall scoring metric, the tool provides users with an informed decision on the safest route. Through the integration of advanced AI algorithms and the incorporation of various data sources, this robust navigation tool significantly enhances the accuracy and reliability of route selection, ultimately improving overall road safety and enabling users to make well-informed decisions for their journeys. This distinctive contribution highlights the potential of AI-driven technologies to revolutionize the field of navigation and enhance road user safety.

Although the current tool can be considered as a starting point, there remains a need for a more robust safe navigation tool. Table 6 summarizes future research needs.

Table 6. Future Research Needs

Problem	Research Topic
Accurate crash prediction models	The safe route-finding algorithm should be designed as a predictive algorithm to account for changes in the risk of being involved in crashes. Future research is required to improve the accuracy of prediction models.
Disaggregated crash prediction models for measuring safety	Develop vehicle-level crash prediction models by providing the users with modified information about their trips based on the user's driving style, historical record of crashes, potential weaknesses, etc.
Short duration crash prediction models	Develop short-duration (e.g., daily, hourly) crash prediction models to provide more real-time aspects of safety on the navigation tools.
Measurement of route safety	Introduce new methods to aggregate road segment and vehicle-level risks at the route level.
Collecting crash and traffic data in real-time to support dynamic, predictive route-finding	Investigate the potential of using crowdsourced data in safe route-finding algorithms.
Trade-off between safety and time	Investigate the sensitivity of users to the risk of being involved in crashes and travel time.
Centralized or decentralized navigation system	Explore the potential safety impacts of two types of safe route-finding algorithms: centralized and decentralized.

Additional Products

The Education and Workforce Development (EWD) and Technology Transfer (T2) products created as part of this project are described below and are listed on the SafeD Website. The final project dataset is located on the SafeD Dataverse[1].

Education and Workforce Development Products

Undergraduate and graduate courses:

- Texas State University CE 4361 Highway Engineering.
- Three-hour long workshop at National Summer Transportation Institute (NSTI), Texas State University on July 20, 2023.
- UTC presentations

Student Funding and Enrichment:

- TTI – two Ph.D. students, Yanmo Weng and Shoujia Li, at Texas A&M University.
- TTI – one undergraduate student, Valerie Vierkant, at Texas A&M University.

For Yanmo Weng and Shoujia Li, the project has been very beneficial. This project allowed both to enhance their knowledge in safety analysis and statistics, learn new programming languages, and publish papers. Valerie Vierkant learned how to assemble different types of traffic and roadway data, perform data quality control checks, process and analyze data, link databases, and download data from Waymo and the National Highway Traffic Safety Administration.

Technology Transfer Products

The main technology transfer products from this study include the following:

- Datasets
- Webinar
- Li, S., Das, S., Ye, X., and Mills, D. Development of a Safe Route Navigation Tool. ITE Annual Meeting. August 13-16, 2023, Portland, Oregon.
- Journal Article – Sohrabi, S., Weng, Y., Das, S., and Paal, S. Safe route-finding: A review of literature and future directions. Accident Analysis & Prevention. Volume 177, 2022. https://doi.org/10.1016/j.aap.2022.106816

[1] https://doi.org/10.15787/VTT1/AL4C8V

Data Products

The research team uploaded one geodatabase 06_002 (06_002SampleData.dbf, 06_002SampleData.shp, 06_002SampleData.prj, and 06_002SampleData.shx) along with their metadata. The metadata describes the data, including the source, description and coding of categorical variables, and number of missing values.

Acknowledgments

This study was funded by the SafeD University Transportation Center (UTC) project Developing AI-driven Safe Navigation Tool.

References

Abdelhamid, S., Elsayed, S. A., AbuAli, N. & Hassanein, H. S. 2016. Driver-centric route guidance. *IEEE*, 1-6.

Abdelrahman, A., Hassanein, H. S. & Abu-Ali, N. 2019. IRouteSafe: Personalized cloud-based route planning based on risk profiles of drivers. *IEEE*, 1-6.

Alcolea, A., M. E. Paoletti, J. M. Haut, J. Resano, and A. Plaza. 2020. Inference in Supervised Spectral Classifiers for On-Board Hyperspectral Imaging: An Overview. *Remote Sensing*, Vol. 12, No. 3, p. 534. https://doi.org/10.3390/rs12030534.

Alpkoçak, A. & Cetin, A. Safe Map Routing Using Heuristic Algorithm Based on Regional Crime Rates. 2019 2020. *Springe*r, 335-346.

Amin-Naseri, M., Chakraborty, P., Sharma, A., Gilbert, S. B. & Hong, M. 2018. Evaluating the reliability, coverage, and added value of crowdsourced traffic incident reports from Waze. *Transportation research record,* 2672**,** 34-43.

Arksey, H. & O'Malley, L. 2005. Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology,* 8**,** 19-32.

Auer, A., Feese, S., Lockwood, S. & Hamilton, B. A. 2016. History of intelligent transportation systems. United States. Department of Transportation. *Intelligent Transportation ….*

Bao, S., Nitta, T., Yanagisawa, M. & Togawa, N. 2017. A Safe and Comprehensive Route Finding Algorithm for Pedestrians Based on Lighting and Landmark Conditions. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* E100.A**,** 2439-2450.

Basso, F., Pezoa, R., Varas, M. & Villalobos, M. 2021. A deep learning approach for real-time crash prediction using vehicle-by-vehicle data. *Accident Analysis and Prevention,* 162**,** 106409.

Bing Maps Dev Center - Bing Maps Dev Center [WWW Document], n.d. URL https://www.bingmapsportal.com/ (accessed 5.29.23).

Bray, H. 2014. You are here: from the compass to GPS, the history and future of how we find ourselves, *Basic Books (AZ)*.

Breiman, L. 2001. Random Forests. *Machine Learning*, Vol. 45, No. 1, pp. 5–32. https://doi.org/10.1023/A:1010933404324.

Bura, D., Singh, M. & Nandal, P. 2019. Predicting Secure and Safe Route for Women using Google Maps. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 14-16 Feb. 103-108.

Byon, Y.-J., Abdulhai, B. & Shalaby, A. 2010. Incorporating Scenic View, Slope, and Crime Rate into Route Choices: Emphasis on Three-Dimensional Geographic Information Systems with Digital Elevation Models and Crime Rate Geospatial Data. *Transportation research record,* 2183**,** 94-102.

Canales, J. 2018. Navigating the history of GPS. *Nature Publishing Group*.

Cantarero, R., Rubio, A., Romero, M. J. S., Dorado, J., Fernández, J. & Lopez, J. C. 2021. COVID19-Routes: A Safe Pedestrian Navigation Service. *IEEE Access,* 9**,** 93433-93449.

Chandra, S. 2014. Safety-based path finding in urban areas for older drivers and bicyclists. *Transportation research part C: emerging technologies,* 48**,** 143-157.

Chen, N. & Lou, B. N. 2021. A self-driving tour service system based on traffic safety and accessibility. *Advances in transportation studies,* 53.

Cortes, C., and V. Vapnik. 1995. Support-Vector Networks. *Machine Learning*, Vol. 20, No. 3, pp. 273–297. https://doi.org/10.1007/BF00994018.

de Leur, P. & Sayed, T. 2003. A framework to proactively consider road safety within the road planning process. *Canadian Journal of Civil Engineering*, 30, 711-719.

De'ath, G. 2007. Boosted Trees for Ecological Modeling and Prediction. *Ecology*, Vol. 88, No. 1, pp. 243–251. https://doi.org/10.1890/0012-9658(2007)88[243:BTFEMA]2.0.CO;2.

Deng, Y., Chen, Y., Zhang, Y., Mahadevan, S., 2012. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing* 12, 1231–1237. https://doi.org/10.1016/j.asoc.2011.11.011

Ding, Y., Chen, C., Zhang, S., Guo, B., Yu, Z. & Wang, Y. 2017. Greenplanner: Planning personalized fuel-efficient driving routes using multi-sourced urban data. *IEEE*, 207-216.

Dong, W. 2011. An overview of in-vehicle route guidance system. *Australasian Transport Research Forum.*

Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., Jurišica, L., 2014. Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Engineering* 96, 59–69. https://doi.org/10.1016/j.proeng.2014.12.098

El-Wakeel, A. S., Noureldin, A., Hassanein, H. S. & Zorba, N. 2018. IDriveSense: Dynamic route planning involving roads quality information. *IEEE,* 1-6.

Eren, E. & Tuzkaya, U. R. 2021. Safe distance-based vehicle routing problem: Medical waste collection case study in COVID-19 pandemic. *Computers & Industrial Engineering,* 157**,** 107328.

Friedman, J. H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, Vol. 29, No. 5, pp. 1189–1232. https://doi.org/10.1214/aos/1013203451.

Friedman, J. H. 2002. Stochastic Gradient Boosting. *Computational Statistics & Data Analysis*, Vol. 38, No. 4, pp. 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2.

Friedman, J. H., and J. J. Meulman. 2003. Multiple Additive Regression Trees with Application in Epidemiology. *Statistics in Medicine*, Vol. 22, No. 9, pp. 1365–1381. https://doi.org/10.1002/sim.1501.

Galbrun, E., Pelechrinis, K. & Terzi, E. 2016. Urban navigation beyond shortest route: The case of safe paths. *Information Systems,* 57**,** 160-171.

Gallo, G. & Pallottino, S. 1988. Shortest path algorithms. *Annals of Operations Research,* 13**,** 1-79.

Hastie, T., R. Tibshirani, and J. H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hayes, S., Wang, S. & Djahel, S. 2020. Personalized Road Networks Routing with Road Safety Consideration: A Case Study in Manchester. *IEEE*, 1-6.

He, Z. & Qin, X. 2017. Incorporating a safety index into pathfinding. *Transportation Research Record*, 2659, 63-70.

Hefner, J. T., M. K. Spradley, and B. Anderson. 2014. Ancestry Assessment Using Random Forest Modeling. *Journal of Forensic Sciences*, Vol. 59, No. 3, pp. 583–589. https://doi.org/10.1111/1556-4029.12402.

Herbert, W. & Mili, F. 2008. Route guidance: state of the art vs. state of the practice. *2008 IEEE Intelligent Vehicles Symposium*. IEEE, 1167-1174.

Holden, J., Reinicke, N. & Cappellucci, J. 2020. RouteE: A Vehicle Energy Consumption Prediction Engine. *Society of Automotive Engineers Technical Paper Series,* 2.

Hoseinzadeh, N., Arvin, R., Khattak, A. J. & Han, L. D. 2020. Integrating safety and mobility for pathfinding using big data generated by connected vehicles. *Journal of Intelligent Transportation Systems,* 24**,** 404-420.

Hossain, M., Abdel-Aty, M., Quddus, M. A., Muromachi, Y. & Sadeek, S. N. 2019. Real-time crash prediction models: State-of-the-art, design pathways and ubiquitous requirements. *Accident Analysis & Prevention,* 124**,** 66-84.

Ito, S. & Koji, Z. 2020. Assessing a risk-avoidance navigation system based on localized torrential rain data. *EDP Sciences*, 03006.

Jones Live Map Meter Company. (1910). *The Jones Live Map: What Happens Without It.*

Kamal Alsheref, F. 2019. Route Recommendation Model Via An Analytic Hierarchy Process (AHP). *Journal of Advanced Research in Dynamical and Control Systems,* 11**,** 17-24.

Kaur, R., Goyal, V., Guntur, V. M. V., Saini, A., Sanadhya, K., Gupta, R. & Ratra, S. A. 2021. Navigation System for Safe Routing. *IEEE,* 240-243.

Khanfor, A., Friji, H., Ghazzai, H. & Massoud, Y. 2020. A Social IoT-Driven Pedestrian Routing Approach During Epidemic Time. *2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, 12-16 Dec. 1-6.

Khanjary, M. & Hashemi, S. M. 2012. Route guidance systems: review and classification. 2012. *2012 6th Euro American Conference on Telematics and Information Systems (EATIS)*, Valencia, Spain. IEEE, 1-7.

Krumm, J. & Horvitz, E. 2017. Risk-aware planning: Methods and case study for safer driving routes.

Kusano, S. & Inoue, M. 2013. Safety route guidance system using participatory sensing. *IEEE*, 363-364.

Levy, S., Xiong, W., Belding, E. & Wang, W. Y. 2020. SafeRoute: Learning to Navigate Streets Safely in an Urban Environment. *ACM Transactions on Intelligent Systems and Technology (TIST),* 11**,** 1-17.

Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., Filev, D. & Michelini, J. Cloud aided safety-based route planning. 2014 2014. *IEEE*, 2495-2500.

Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., Filev, D. P. & Michelini, J. 2016. Road Risk Modeling and Cloud-Aided Safety-Based Route Planning. *IEEE Transactions on Cybernetics*, 46, 2473-2483.

Liaw, A. and M. Wiener. 2002. Classification and Regression by RandomForest. R news, Vol. 2, No. 3, pp. 18–22.

Lin, Y. & Li, R. 2020. Real-time traffic accidents post-impact prediction: Based on crowdsourcing data. *Accident Analysis & Prevention,* 145**,** 105696.

Liu, Q., Kumar, S. & Mago, V. SafeRNet: Safe transportation routing in the era of Internet of vehicles and mobile crowd sensing. 2017. *14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 8-11 Jan. 2017 2017. 299-304.

Lord, D., Qin, X. & Geedipally, S. 2021. *Highway safety analytics and modeling*, Elsevier.

Lord, D., Washington, S. P. & Ivan, J. N. 2005. Poisson, Poisson-gamma and zero-inflated regression models of motor vehicle crashes: balancing statistical fit and theory. *Accident* Analysis and Prevention, 37, 35-46.

Lozano Domínguez, J. M. & Mateo Sanguino, T. d. J. 2021. Walking Secure: Safe Routing Planning Algorithm and Pedestrian's Crossing Intention Detector Based on Fuzzy Logic App. *Sensors*, 21, 529.

M. Noto, H. Sato, 2000. A method for the shortest path search by extended Dijkstra algorithm. Presented at the Smc 2000 conference proceedings. *2000 IEEE international conference on systems, man and cybernetics. "cybernetics evolving to systems, humans, organizations, and their complex interactions"* (cat. no.0, pp. 2316–2320 vol.3.) https://doi.org/10.1109/ICSMC.2000.886462

Mannering, F. 2018. Temporal instability and the analysis of highway accident data. *Analytic methods in accident research*, 17, 1-13.

Mata, F., Torres-Ruiz, M., Guzmán, G., Quintero, R., Zagal-Flores, R., Moreno-Ibarra, M. & Loza, E. 2016. A mobile information system based on crowd-sensed and official crime data for finding safe routes: A case study of Mexico City. *Mobile Information Systems*, 2016.

Mehta, H., Kanani, P., & Lande, P. 2019. Google maps. Int. J. Comput. Appl 178, 41–46.

Mishra, S., Singh, N. & Bhattacharya, D. 2021. Application-Based COVID-19 Micro-Mobility Solution for Safe and Smart Navigation in Pandemics. *ISPRS International Journal of Geo-Information*, 10, 571.

Munn, Z. & Peters, M. J, Stern, C, Tufanaru, C, McArthur, A, & Aromataris, E. 2018. Systematic review or scoping review? Guidance for authors when choosing between a systematic or scoping review approach. *BMC Medical Research Methodology,* 18, 1-7.

Official MapQuest - Maps, Driving Directions, Live Traffic [WWW Document], n.d. URL https://www.mapquest.com/ (accessed 5.29.23).

Ouyang, W., Yu, C.-W., Yu, K.-M., Lin, K.-J. & Chang, H.-T. 2014. Safe path planning strategy for bike net. *Wireless personal communications,* 78, 1995-2007.

Özdoğan-Sarıkoç, G., M. Sarıkoç, M. Celik, and F. Dadaser-Celik. 2023. Reservoir Volume Forecasting Using Artificial Intelligence-Based Models: Artificial Neural Networks, Support Vector Regression, and Long Short-Term Memory. *Journal of Hydrology*, Vol. 616, p. 128766. https://doi.org/10.1016/j.jhydrol.2022.128766.

Petridou, E. & Moustaki, M. 2000. Human factors in the causation of road traffic crashes. *European journal of epidemiology,* 16**,** 819-826.

Preda, A., Rönkkö, M., Pickl, S. & Kolehmainen, M. 2013. GIS-based route planning for HAZMAT transportation. *Springer*, 357-366.

Prokhorenkova, L., G. Gusev, A. Vorobev, A. V. Dorogush, & A. Gulin. 2018. CatBoost: Unbiased Boosting with Categorical Features. Red Hook, NY, USA.

Puthige, I., Bansal, K., Bindra, C., Kapur, M., Singh, D., Mishra, V. K., Aggarwal, A., Lee, J., Kang, B.-G. & Nam, Y. 2021. Safest Route Detection via Danger Index Calculation and K-Means Clustering.

Qiu, L. & Nixon, W. A. 2008. Effects of adverse weather on traffic crashes: systematic review and meta-analysis. *Transportation Research Record,* 2055**,** 139-146.

Radojičić, N., Marić, M. & Takači, A. 2018. A new fuzzy version of the risk-constrained Cash-in-Transit Vehicle Routing Problem. *Information Technology and Control/Informacinės technologijos ir valdymas,* 47**,** 321-337.

Santhanavanich, T., Wuerstle, P., Silberer, J., Loidl, V., Rodrigues, P. & Coors, V. 2020. 3D safe routing navigation application for pedestrians and cyclists based on open source tools. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences,* 6.

Sargeant, J. M. & O'Connor, A. M. 2020. Scoping reviews, systematic reviews, and meta-analysis: applications in veterinary medicine. *Frontiers in veterinary science,* 7**,** 11.

Sarraf, R. & McGuire, M. P. 2018. A data driven approach for safe route planning. *International Journal of Applied Geospatial Research (IJAGR),* 9**,** 1-18.

Schmitt, E. & Jula, H. Vehicle Route Guidance Systems: Classification and Comparison. 2006. *IEEE Intelligent Transportation Systems Conference*, 17-20 Sept. 2006 2006. 242-247.

Shah, M., Liu, T., Chauhan, S., Qi, L. & Zhang, X. 2020. CycleSafe: Safe Route Planning for Urban Cyclists.

Shubenkova, K., Boyko, A. & Buyvol, P. 2018. The technique of choosing a safe route as an element of smart mobility. *Transportation research procedia,* 36**,** 718-724.

Smola, A. J., and B. Schölkopf. A. 2004. Tutorial on Support Vector Regression. *Statistics and Computing*, Vol. 14, No. 3, pp. 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

Sohrabi, S. & Lord, D. 2022. Navigating to safety: Necessity, requirements, and barriers to considering safety in route finding. *Transportation Research Part C: Emerging Technologies,* 137**,** 103542.

Soni, S., Shankar, V. G. & Chaurasia, S. 2019. Route-the safe: A robust model for safest route prediction using crime and accidental data. *Int. J. Adv. Sci. Technol,* 28**,** 1415-1428.

Takeno, R., Seki, Y., Sano, M., Matsuura, K., Ohira, K. & Ueta, T. 2016. A route navigation system for reducing risk of traffic accidents. *IEEE*, 1-5.

Tarko, A. P., Guo, Q. & Pineda-Mendez, R. 2021. Using Emerging and Extraordinary Data Sources to Improve Traffic Safety. 2326-6325*,* Available: <https://doi.org/10.5703/1288284317283>

Tricco, A. C., Lillie, E., Zarin, W., O'Brien, K. K., Colquhoun, H., Levac, D., Moher, D., Peters, M. D. J., Horsley, T., Weeks, L., Hempel, S., Akl, E. A., Chang, C., McGowan, J., Stewart, L., Hartling, L., Aldcroft, A., Wilson, M. G., Garritty, C., Lewin, S., Godfrey, C. M., Macdonald, M. T., Langlois, E. V., Soares-Weiser, K., Moriarty, J., Clifford, T., Tuncalp, O. & Straus, S. E.

2018. PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. *Ann Intern Med,* 169**,** 467-473.

Utamima, A. & Djunaidy, A. 2017. Be-safe travel, a web-based geographic application to explore safe-route in an area. August 01, 020023.

Vapnik, V. N. 2000. The Nature of Statistical Learning Theory. *Springer*, New York, NY.

Washington, S., Karlaftis, M., Mannering, F. & Anastasopoulos, P. 2020. *Statistical and econometric methods for transportation data analysis*, Chapman and Hall/CRC.

Xu, R., & F. Luo. 2021. Risk Prediction and Early Warning for Air Traffic Controllers' Unsafe Acts Using Association Rule Mining and Random Forest. *Safety Science*, Vol. 135, p. 105125. https://doi.org/10.1016/j.ssci.2020.105125.

Yang, T., A. A. Asanjan, E. Welles, X. Gao, S. Sorooshian, & X. Liu. 2017. Developing Reservoir Monthly Inflow Forecasts Using Artificial Intelligence and Climate Phenomenon Information. *Water Resources Research*, Vol. 53, No. 4, pp. 2786–2812. https://doi.org/10.1002/2017WR020482.

Yew, K. H., Ha, T. T. & Paua, S. D. S. J. 2010. SafeJourney: A pedestrian map using safety annotation for route determination. IEEE, 1376-1381.

Zanakis, S. H., Solomon, A., Wishart, N. & Dublish, S. 1998. Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research,* 107**,** 507-529.

Zeng, W., Miwa, T. & Morikawa, T. 2017. Application of the support vector machine and heuristic k-shortest path algorithm to determine the most eco-friendly path with a travel time constraint. *Transportation Research Part D: Transport and Environment,* 57**,** 458-473.

Zhang, T., W. Lin, A. M. Vogelmann, M. Zhang, S. Xie, Y. Qin, & J.-C. Golaz. 2021. Improving Convection Trigger Functions in Deep Convective Parameterization Schemes Using Machine Learning. *Journal of Advances in Modeling Earth Systems*, Vol. 13, No. 5, p. e2020MS002365. https://doi.org/10.1029/2020MS002365.

Zhou, E., Mao, S. & Li, M. 2017. Investigating street accident characteristics and optimal safe route recommendation: A case study of New York City. *2017 25th International Conference on Geoinformatics*, 2-4 Aug. 1-7.

Appendix A: Literature Review

The researchers used a scoping review framework by Arksey and O'Malley (2005) to explore the literature on safe route-finding. Scoping reviews have a wider scope and more inclusive criteria than systematic reviews (Munn & Peters, 2018; Sargeant & O'Connor, 2020). Their goal is to map the literature, identify key concepts, gaps, and limitations, and determine if a systematic review is needed. Unlike systematic reviews, scoping reviews do not assess the quality or aggregate the literature (Munn & Peters, 2018). This study focused on the exploration of the algorithms and approaches employed in safe route-finding.

A search strategy was formulated to acquire relevant research evidence from three electronic research databases—Scopus, Web of Science, and Institute of Electrical and Electronics Engineers (IEEE) Xplore, along with the reference lists of the retrieved publications. A thorough search was conducted across databases to locate published and indexed articles, letters, reports, book chapters, and books that incorporated a variety of keywords within their titles, abstracts, and keywords. Table 7 summarizes the two sets of keywords. The review included materials published up until October 2021.

Table 7. Keywords Summary

Category	Keywords
Route-finding	Navigation system; Routing; Phone navigation; Car navigation; Bicycle navigation; Vehicle navigation; Automobile navigation; Pedestrian navigation; Road navigation; Route finding; Path finding; Road finding; Road guidance; Route guidance; Vehicle guidance; Car guidance; Navigation device; Route planning; Urban navigation; Vehicle information system; Navigation device
Safety	Safe; Crash; Collision; Accident; Crime; Hazard

After reviewing the titles and abstracts of the identified records, the full text was examined based on predefined inclusion criteria designed to effectively address the review questions. Three distinct criteria were established for this purpose.

1. Must discuss safety in route guidance as a system. Studies that only discuss route-finding algorithms are not included.
2. Must investigate route-finding in land transportation rather than in aviation and maritime transport.
3. Must propose an algorithm or quantification method instead of being a commentary publication.

In total, the search yielded 5,955 publications, with 2,720, 2,246, and 989 publications sourced from Scopus, the Web of Science, and IEEE Xplore, respectively. Out of these, 40 publications were identified that met the inclusion criteria and thus underwent review. The PRISMA scoping review flow diagram (Tricco et al., 2018) in Figure 4 provides an overview of the implemented scoping review. Figure 5 illustrates the yearly publication count starting from 2010. Notably, there has been an increase in publications, indicating a growing research interest in safe route-finding in recent years. The results of this review are discussed in the following sections. The literature covering the definition of safety, the measurement of safety, and safe route-finding algorithms are all summarized (Table 8 shows overall summaries of the literature).

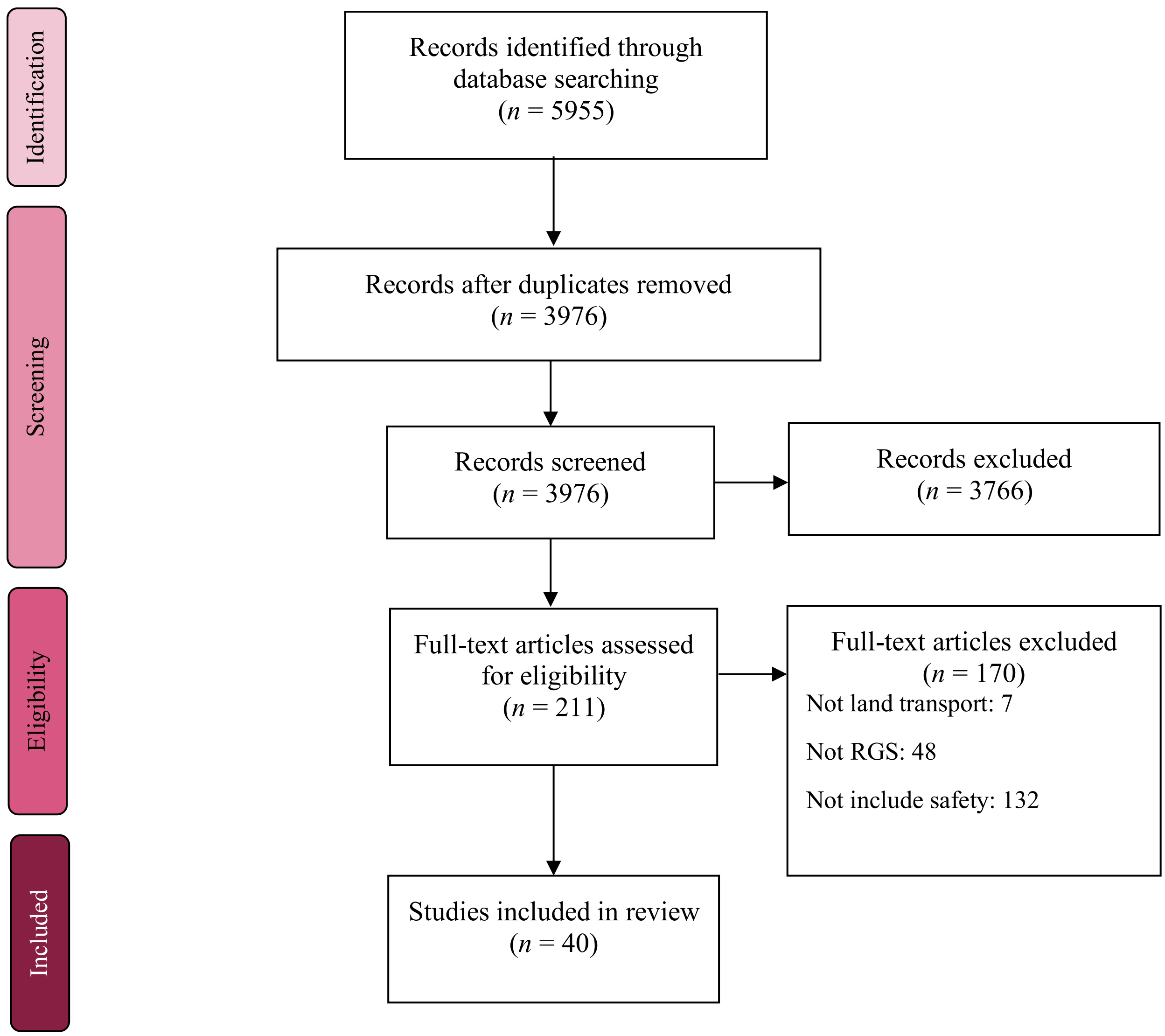


Figure 4. Study identification and selection mechanism of the implemented scoping review.

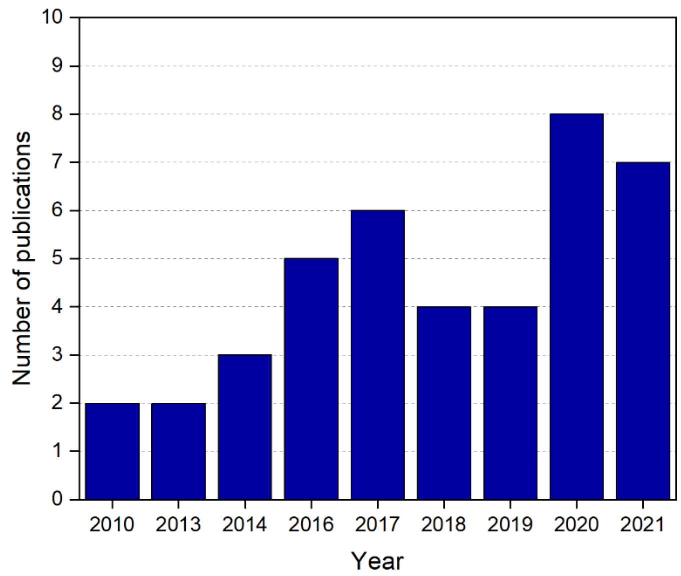


Figure 5. Publications by year (as of October 2021).

Table 8. A Summary of the Reviewed Studies

Study	Definition of Safety	Measurement of Safety	Algorithm: Predictive/ Reactive	Algorithm: Dynamic/ Static	Algorithm: Centralized/ Decentralized
(Santhanavanich et al., 2020)	Pedestrian or cyclist crash risk	Data-driven	Reactive	Dynamic	decentralized
(Sarraf & McGuire, 2018)	Vehicle crash risk	Scoring	Predictive	Static	decentralized
(Mata et al., 2016)	Crime risk	Data-driven	Predictive	Dynamic	decentralized
(Kaur et al., 2021)	Crime risk	Scoring	Reactive	Dynamic	decentralized
(Radojičić et al., 2018)	Vehicle crash risk	Data-driven	Predictive	Static	decentralized
(Takeno et al., 2016)	Vehicle crash risk	Data-driven	Predictive	Static	decentralized
(Bao et al., 2017)	Pedestrian or cyclist crash risk	Scoring	Reactive	Static	decentralized
(Chen & Lou, 2021)	Vehicle crash risk	Data-driven	Predictive	Static	decentralized
(Khanfor et al., 2020)	Health risk	Safety indicator	Predictive	Static	decentralized
(Mishra et al., 2021)	Health risk	Scoring	Predictive	Static	decentralized
(Ito & Koji, 2020)	Vehicle crash risk	Scoring	Reactive	Dynamic	decentralized
(Utamima & Djunaidy, 2017)	Crime risk	Data-driven	Reactive	Static	decentralized

Study	Definition of Safety	Measurement of Safety	Algorithm: Predictive/ Reactive	Algorithm: Dynamic/ Static	Algorithm: Centralized/ Decentralized
(Li et al., 2014)	Vehicle crash risk	Data-driven	Reactive	Dynamic	decentralized
(Cantarero et al., 2021)	Health risk	Safety indicator	Predictive	Dynamic	decentralized
(Shah et al., 2020)	Pedestrian or cyclist crash risk	Scoring	Reactive	Dynamic	decentralized
(Abdelhamid et al., 2016)	Vehicle crash risk	Data-driven	Predictive	Dynamic	decentralized
(Preda et al., 2013)	HAZMAT risk	Scoring	Predictive	Static	decentralized
(El-Wakeel et al., 2018)	Vehicle crash risk	Safety indicator	Reactive	Dynamic	decentralized
(He & Qin, 2017)	Vehicle crash risk	Data-driven	Predictive	Static	decentralized
(Byon et al., 2010)	Crime risk	Scoring	Predictive	Static	decentralized
(Hoseinzadeh et al., 2020)	Vehicle crash risk	Data-driven	Predictive	Dynamic	decentralized
(Zhou et al., 2017)	Vehicle crash risk	Data-driven	Predictive	Static	decentralized
(Abdelrahman et al., 2019)	Vehicle crash risk	Data-driven	Predictive	Dynamic	decentralized
(Hayes et al., 2020)	Vehicle crash risk	Data-driven	Predictive	Static	decentralized
(Bura et al., 2019)	Crime risk	Data-driven	Predictive	Static	decentralized
(Krumm & Horvitz, 2017)	Vehicle crash risk	Data-driven	Predictive	Static	decentralized
(Li et al., 2016)	Vehicle crash risk	Data-driven	Predictive	Dynamic	decentralized
(Kamal Alsheref, 2019)	Vehicle crash risk	Scoring	Predictive	Static	decentralized
(Soni et al., 2019)	Vehicle crash risk/Crime risk	Scoring	Predictive	Static	decentralized
(Eren & Tuzkaya, 2021)	HAZMAT risk	Scoring	Predictive	Static	decentralized
(Alpkoçak & Cetin, 2020)	Crime risk	Data-driven	Predictive	Static	decentralized
(Ouyang et al., 2014)	Pedestrian or cyclist crash risk	Safety indicator	Predictive	Dynamic	decentralized
(Yew et al., 2010)	Pedestrian or cyclist crash risk	Data-driven	Predictive	Static	decentralized

Study	Definition of Safety	Measurement of Safety	Algorithm: Predictive/ Reactive	Algorithm: Dynamic/ Static	Algorithm: Centralized/ Decentralized
(Liu et al., 2017)	Vehicle crash risk	Data-driven	Predictive	Dynamic	decentralized
(Levy et al., 2020)	Crime risk	Data-driven	Predictive	Static	decentralized
(Puthige et al., 2021)	Vehicle crash risk/Crime risk	Data-driven	Predictive	Static	decentralized
(Chandra, 2014)	Pedestrian or cyclist crash risk	Data-driven	Predictive	Static	decentralized
(Lozano Domínguez & Mateo Sanguino, 2021)	Pedestrian or cyclist crash risk	Safety indicator	Predictive	Static	decentralized
(Shubenkova et al., 2018)	Pedestrian or cyclist crash risk	Safety indicator	Predictive	Static	decentralized
(Galbrun et al., 2016)	Crime risk	Data-driven	Predictive	Static	decentralized

Appendix B: Data Preparation for Texas-based Case Study

The following steps were conducted to prepare the databases for analysis:

- First, the datasets from CRIS and TxDOT (w/o assets) were loaded into the R environment. The CRIS dataset was converted to a point shapefile from its original CSV format using the "st_as_sf" function from the "SF" package". The CRIS dataset was assigned the same geographic coordinate system as the Roadway Inventory dataset, North American Datum 1983. The "st_nearest_feature" function from the "SF" package was used to create an index of the closest road, from the TxDOT dataset, in proximity to each CRIS recorded crash. The "st_nearest_feature" function is a pairwise function that determines the nearest points between pairs of geometries. The function compares the latitude and longitude of each crash point to the latitudes and longitudes of every polyline road segment. The algorithm is considered a Brute Force algorithm which has high computational requirements. There is no need to explore the use case of a faster algorithm at the potential cost of data quality because this algorithm was only used once as a data preparation step and thus has no effect on the computational performance of the end-user experience. The relevant code for this data pre-processing step is in Figure 6.
- Next, a sum of the number of crashes at each segment was determined based on the crash index created by the "st_nearest_feature" function. The sum for each segment was added to the TxDOT Roadway Inventory dataset in a newly created field, "numCrashes". Of the 514,480 roadway segments in the TxDOT w/o Assets dataset, there were 64,684 roads with crashes. The file was reduced to only the 64,684 road segments on which crashes occurred prior to normalizing the data. The file reduction resulted in a 97.71% faster computation time for normalizing the data.
- The lengths of road segments in the TxDOT dataset are not equal and have a large variation. The length of road segments with crashes ranges from approximately 23 feet to 635 miles. To better quantify the risk associated with the number of crashes on each road segment, the length of the road segment is a necessary factor to include, as a 1,000-foot-long road segment with 50 crashes is more dangerous than a 100,000-foot-long road segment with 500 crashes.

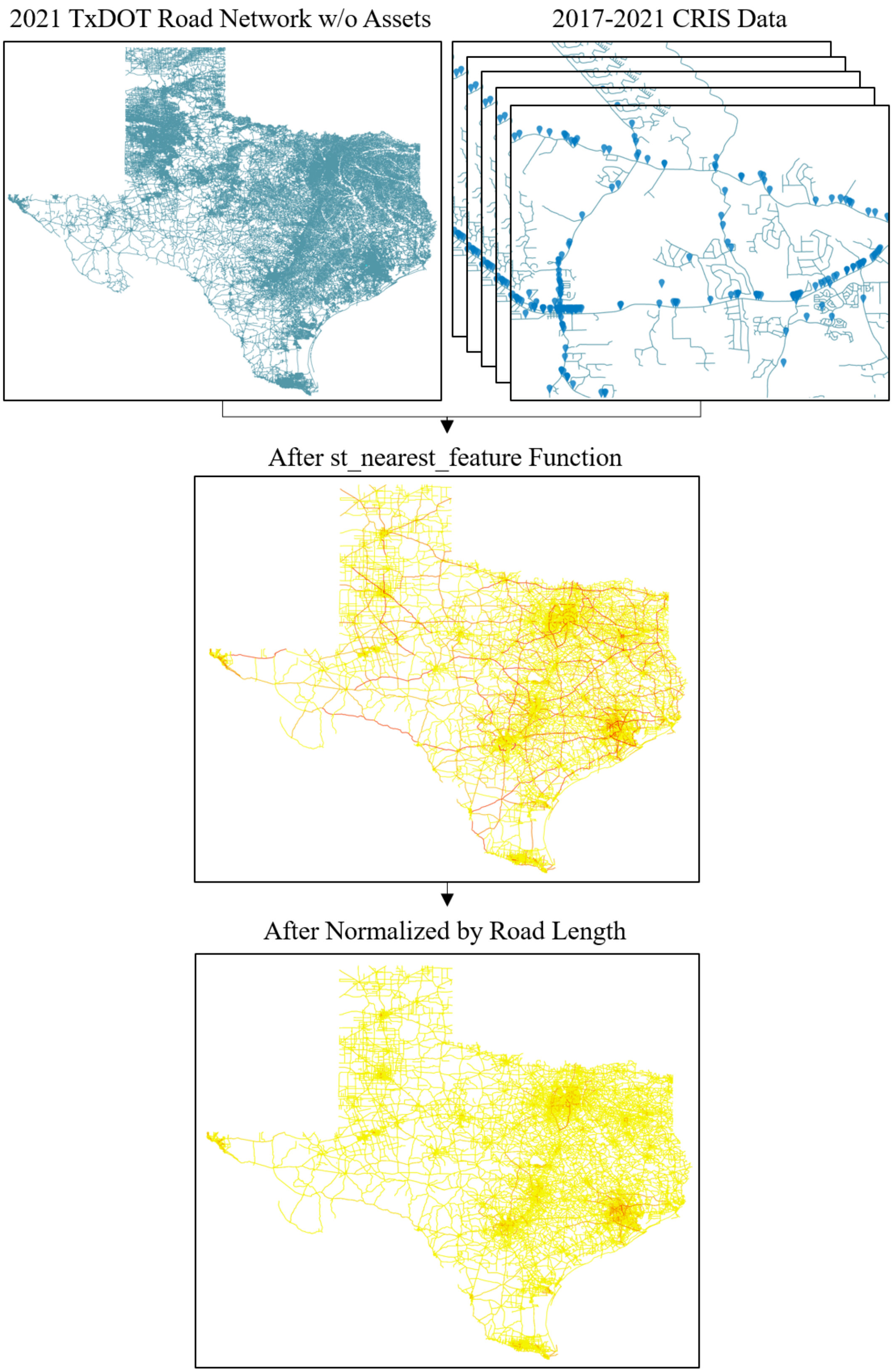


Figure 6. Assigning traffic crash data on RHiNO network.

- Prior to normalizing the data, the length of the road segments was calculated into miles using the "st_length" function and the geometry of each polyline. Then, the number of crashes was divided by the segment length to create a common metric across the dataset representing the number of crashes per mile of road. The crashes per mile of road segment was then normalized on a scale of 0 (no crashes/mile) to 1 (the highest number of crashes/mile within the dataset) by dividing the crashes per mile for each road segment by the highest number of crashes per mile in the road segment dataset. An example of the difference between risk scores can be seen below in Figure 7a and Figure 7b.

(a) Road segment with the most crashes. 10,120 crashes on an approximately 371-mile-long road segment. A 0.009 risk score on a normalized scale.

(b) Road Segment with the most crashes per foot. 27 crashes on an approximately 49-foot-long road segment. A 1.000 risk score on a normalized scale.

Figure 7. Assigning risk scores.

Appendix C: Artificial Intelligence-based Models

Random Forest (RF)

Random Forest (RF) is a supervised machine learning algorithm that uses bootstrap sampling to create new training sample sets. It builds decision trees based on these samples (Hefner et al., 2014), with each tree being independent (Liaw & Wiener, 2002). RF is based on a decision tree and bagging framework. RF does not prune the decision trees during development. Although individual tree accuracy may be low, combining their predictions generates a more accurate overall result (Xu & Luo, 2021). The RF approach involves two steps: forest formation and decision-making. The algorithm randomly divides training samples into subsets, constructs Classification and Regression Tree (CART) decision trees, and determines predictions using a simple voting method. The final prediction is the average of the predictions from multiple decision trees. See Figure 8 for a visual representation of the RF algorithm.

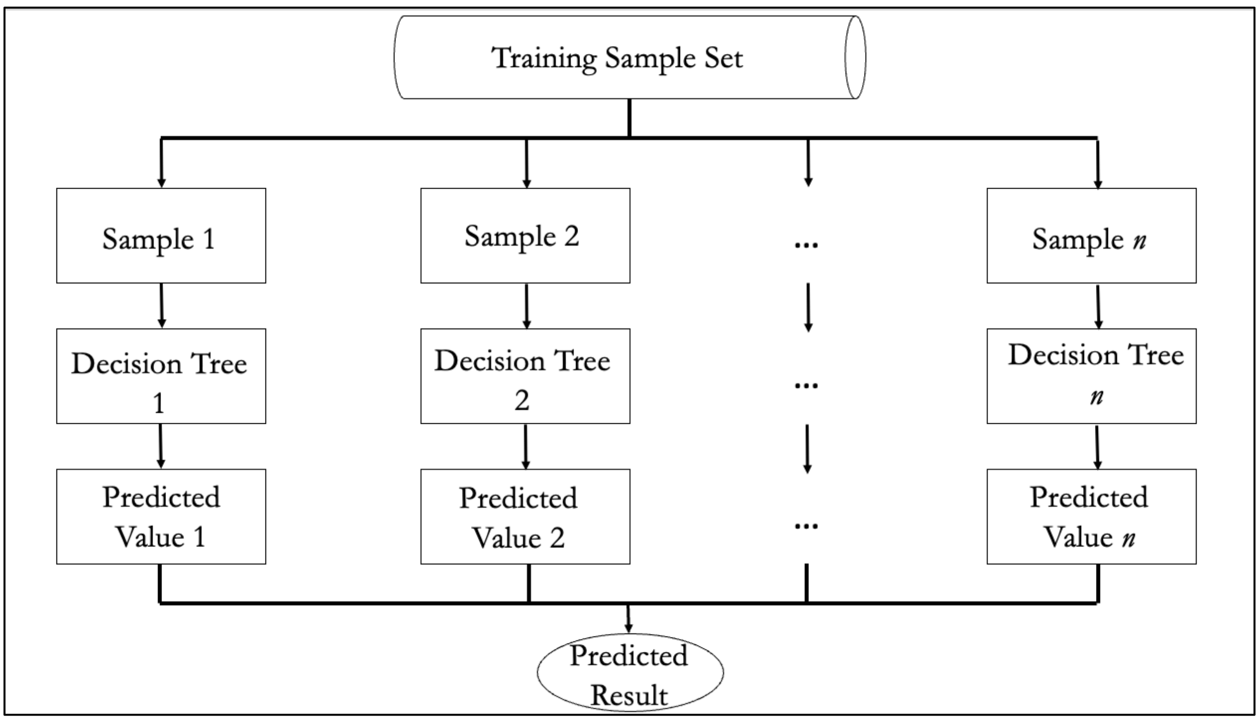


Figure 8. Random Forest algorithm (Xu & Luo, 2021).

The margin function in RF is given by Equation 1 (Breiman, 2001).

$$\mathbf{mg(X, Y) = av_k\, I(h_k\,(X) = Y) - \max_{j \neq Y} av_k\, I(h_k\,(X) = j)} \quad (1)$$

Where h_1 (x), h_2 (x), . . ., h_k (x) are ensembles of classifiers, X, and Y are random vectors, and I is the indicator function.

The margin function evaluates the difference in the average number of votes between the right class that exceeds and the average vote for all other classes. A larger margin indicates a higher level of confidence in the classification (Breiman, 2001). The error generalization is given by:

$$\mathbf{PE}^{*} = \mathbf{P}_{\mathbf{X},\mathbf{Y}}\left(\mathbf{mg}(\mathbf{X},\mathbf{Y}) < \mathbf{0}\right) \tag{2}$$

Where X, Y indicates that the probability is over the X, Y space.

Gradient Boosting (GB)

The Gradient Boosting (GB) method, also known as multiple additive trees (MAT), is an improvement on Decision Trees (DT) proposed by Friedman (2001, 2003). GB uses stochastic gradient boosting to combine models and boost accuracy in data mining. Boosting enhances learning algorithms by combining low-error-rate models into an ensemble, resulting in improved performance. The flow chart in Figure 9 illustrates the GB machine learning method, where weak classifiers are combined in an ensemble. Incorrectly predicted points are given higher weights in subsequent classifiers, and the final decision is based on the weighted average of individual predictions (Zhang, 2021). A GB model approximates the true functional relationship and can be described by Equation 3 (De'ath, 2007; Hastie et al., 2009).

$$\mathbf{f}(\mathbf{x}) = \sum_{\mathbf{n}} \mathbf{f}_{\mathbf{n}}(\mathbf{x}) = \sum_{\mathbf{n}} \boldsymbol{\beta}_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \boldsymbol{\gamma}_{\mathbf{n}}) \tag{3}$$

Where x is a set of predictors and f(x) is the estimate of the response variable. $g(x, \gamma_n)$ are single decision trees with the parameter γ_n signaling the split variables. Coefficients β_n (n = 1,2,...,n) determine how each tree is joined together. Values of β_n depend on the minimization of a specified loss function, $L(y_i,f(x_i))$. Prediction performance is measured by a loss function, such as deviance. A numerical optimization method named functional gradient descent was proposed by Friedman (2001). The algorithm to initialize $f_0(x)$ is given below.

1. For n=1,2,3,…m (number of trees)
    a. For i = 1 to m (number of observations), calculate the residuals
    $$\tilde{y}_{in} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$$
    b. Fit a decision tree to $\tilde{y}_{in}$ to estimate γ_n
    c. Estimate β_n by minimizing $L(y_i, f_{n-1}(x_i) + \beta_n g(x,\gamma_n))$
    d. Update $f_n(x) = f_{n-1}(x) + \beta_n g(x, \gamma_n)$
2. Calculate
    $$f(x) = \sum_{n} f_n(x)$$

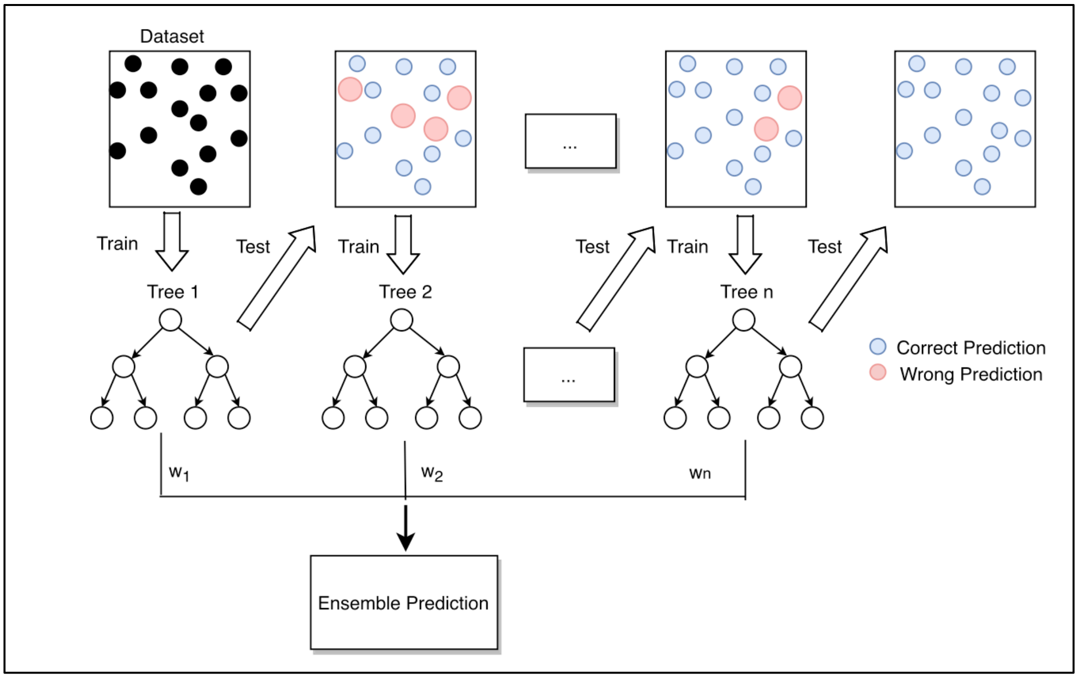


Figure 9. Flowchart for gradient boosting (Zhang et al., 2021).

The iterative tree building process often leads to overfitting, where models perform well only on trained data and have low accuracy on other datasets. To prevent overfitting, the model is tested with a separate dataset. Training stops when the model performs well on both the training and test datasets. Regularization parameters, including learning rate and tree complexity, help address overfitting and improve model performance. The learning rate determines the speed of model improvement, with lower values requiring more data and time but minimizing loss function (De'ath, 2007). Higher values lead to overfitting and poor performance. Tree complexity refers to the number of nodes in a tree, with a simple tree having two nodes and one split (Hastie et al., 2009). Balancing the learning rate and tree complexity is essential to avoid overfitting.

Support Vector Regression (SVR)

Support Vector Regression (SVR) (Cortes & Vapnik, 1995) is a supervised machine learning model derived from Support Vector Machine (SVM) (Vapnik, 2000; Smola & Schölkopf, 2004). SVR is similar to SVM but with a few updates (Yang et al., 2017). It maps the data nonlinearly and solves the linear regression problem in a higher dimensional feature space (see Figure 10). This allows SVR to describe non-linear relationships.

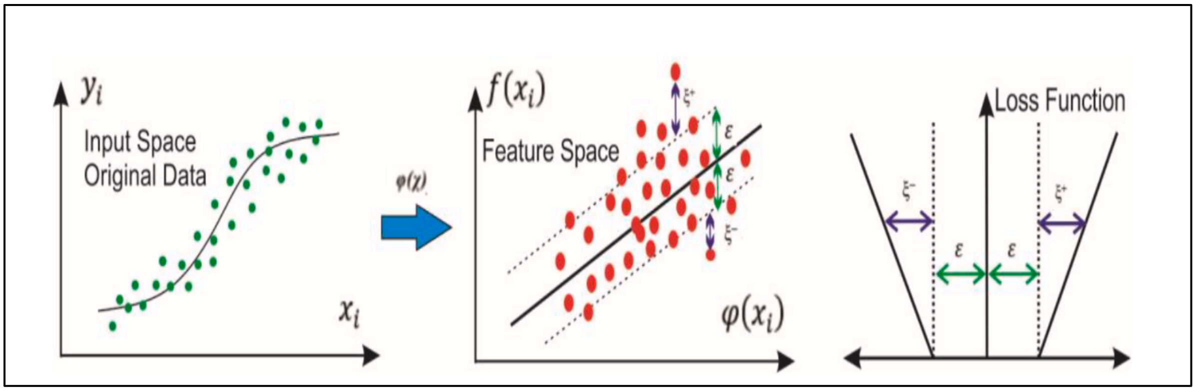


Figure 10. KNN approach using k = 10 (Özdoğan-Sarıkoç et al., 2023).

The regression function is defined by the following equation (Özdoğan-Sarıkoç et al., 2023).

$$\boldsymbol{f(x) \ = \sum_{i=1}^{n}(\partial_i^- - \partial_i^+)\, K\,(x_{i,}x_j) + b} \tag{5}$$

Where $K(x_i,x_j)$ is the kernel function. In this study, the kernel functions tested are radial basis function (RBF), polynomial, and linear kernels, which are shown in Equation (6).

$$\textbf{Polynomial Kernel}: \ \mathbf{K\,(x_i, x_j) \ = \ (\gamma(x.x_i) + b)^d} \tag{6}$$

$$\textbf{Radial Basis Function Kernel} \ = \ \mathbf{K\,(x_i, x_j) \ = \ exp(-\frac{\|x - x_i\|}{2\sigma^2})}$$

Where γ is the structural parameter in the radial basis function, polynomial kernels, ν, represent the residuals, and d is the degree of the polynomial term.

Cat Boosting

CatBoost (CB) is a machine learning approach developed by Yandex engineers in 2017 (Prokhorenkova et al., 2018). It is derived from Gradient Boosting Decision Tree (GBDT) (see Figure 11), a robust technique for solving challenging machine learning problems involving heterogeneous features, noisy data, and complex dependencies.

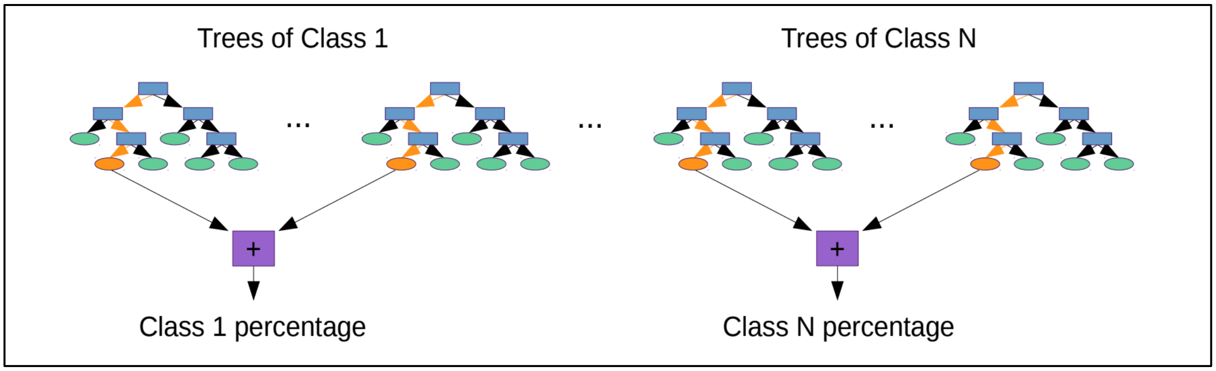


Figure 11. Gradient boosting decision trees (Alcolea et al., 2020).

CatBoost has the following advantages compared to other GBDT algorithms:

Firstly, CB effectively handles categorical data. Traditional GBDT algorithms can replace categorical features with corresponding average label values. In decision trees, the average label value is used for node splitting, referred to as Greedy Target-based Statistics (Greedy TBS), defined by Equation 8 (Prokhorenkova et al., 2018).

$$\frac{\sum_{j=1}^{p}[\mathbf{x_{j,k}} = \mathbf{x_{i,k}}]\mathbf{Y_i}}{\sum_{j=1}^{n}[\mathbf{x_{j,k}} = \mathbf{x_{i,k}}]} \tag{8}$$

When we consider a dataset of observations D={X_i, Y_i} i = 1, ..., n, if a permutation is σ=(σ_1,..., σ_n), $x_{\sigma_{p,k}}$ is substituted with:

$$\frac{\sum_{j=1}^{p=1}\left[x_{\sigma_{j,k}} = x_{\sigma_{j,k}}\right]Y_{\sigma_j} + aP}{\sum_{j=1}^{p=1}\left[x_{\sigma_{j,k}} = x_{\sigma_{j,k}}\right] + a} \tag{9}$$

Where p is a prior value and a is the weight of the prior value. This method contributes to reducing the noise obtained from the low frequency category.

Secondly, CB combines multiple categorical features and their combinations in the current tree with all categorical features in the dataset. Thirdly, CB gradient bias is overcome by CatBoost. GBDT generates a weak learner in each iteration and each learner is trained based on the previous learner's gradient. The output is provided via the accumulation of classified results of all learners (Friedman, 2002). The final learned model might be overfit due to biased pointwise gradient estimation. CB uses a new method, ordered boosting, to change the gradient estimation method in the classic algorithm, which can overcome prediction shifts caused by gradient bias and further enhance the model's generalization ability (Prokhorenkova et al., 2018). CB trains a separate model M_i for each sample X_i to obtain an unbiased gradient estimation. The model M_i is trained with a training set without sample X_i. M_i is used to obtain a gradient estimation of the sample. This gradient is used to train the final model's base learner.