CrashGuard: A BMSSP based Data-Driven and Risk-Aware Routing System for Safer Urban Travel Using Spatial Analytics and Machine Learning

1st Chaithra G V
Dept. of CSE,
Bangalore Institute of Technology
Bengaluru, India
chaithragv@bit-bangalore.edu.in

2nd Riya Singh
Dept. of CSE,
Bangalore Institute of Technology
Bengaluru, India
riyasingh979me@gmail.com

3rd Shalabh Ranjan Prasad
Dept. of CSE,
Bangalore Institute of Technology
Bengaluru, India
ranjanshalabh19@gmail.com

4th Shubh Sinha
Dept. of CSE,
Bangalore Institute of Technology
Bengaluru, India
vsshubh2003@gmail.com

5th Sonal Srivastava
Dept. of CSE,
Bangalore Institute of Technology
Bengaluru, India
sonalshrivastava26@gmail.com

***Abstract*—Due to the rapid growth in vehicular traffic and urban density, road collisions in urban areas, especially metropolitan areas like Bengaluru have become a significant public safety concern. Conventional navigation applications have a greater emphasis on travel distance or travel time, with little assistance related to the safety of routes based on high accident areas and environmental conditions which are evolving consistently. The study introduces a data-driven and safety-conscious routing system that includes spatial analytics, machine learning and real-time weather information in order to inform and recommend the safest route to travel. DBSCAN clustering is deployed to detect "accident-prone" areas from historical data sets, and a BallTree based risk propagation model is then used to create weightings of risk and denote risk across road segments of a network based on OpenStreetMap (OSM) data. Each road segment has a multi-faceted weighting, created from a final composite of distance/normalized distance and risk. Given this weighting, a route is then generated using a Bounded Multi-Source Shortest Path (BMSSP) algorithm. Weather information, obtained through the OpenWeatherMap API, is incorporated to adjust and modify the normalized distance weighting with real-time environmental risk (i.e., rain and fog). The entire framework is deployed through a Streamlit interactive web application and exposes users to real-time route monitoring and individual safety preferences. The system has been experimentally evaluated using Bengaluru traffic datasets, and the proposed framework showcases a strong and 'safer routes' approach without compelling users to sacrifice travel efficiency.**

Keywords— Machine Learning, Accident Risk Prediction, DBSCAN Clustering, Safe Route Computation, Accident Hotspot Detection

I. INTRODUCTION

For many cities, such as Bengaluru, rapid urbanization and the burgeoning population of vehicles on the roads has made traffic safety among the biggest challenges, including accident counts. Bengaluru Traffic Police has provided a map detailing accident spacing across the city, which amounts to thousands of accidents every year and many which turn into significant injuries or fatality. Traditional navigation systems including Google Maps or Waze are designed to help minimize routing distance or travel time, but do not contextualize safety data associated with accidents, hazardous weather or time of day variables for traffic.

Further, with the access of open geospatial data and advancement of spatial data analytics, it is now possible to develop "smart" navigation systems that elicit safety factors with data-driven inputs to identify multiple real-world factors apart from just distance. It may include all types of accident history, spatial clustering algorithms, and environmental risk analysis, etc., to be utilized in navigating and optimizing safety while managing efficiency.

This represents a larger shift from distance and systems thinking to risk that also considers social factors and events affecting risk, which fits into the larger emerging frontier of smart mobility and urban data intelligence.There has been a number of studies that attempt to identify risk zones for accidents using clustering techniques such as K-Means or hierarchical clustering. However, these techniques often expose serious limitations in handling noise and irregular spatial distribution.

In contrast, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is naturally appropriate for geospatial data with the ability to find clusters that are not spherical in shape while disregarding sparse outliers. Furthermore, the use of OpenStreetMap (OSM) graph structures allows for a detailed mapping of the accident hotspots with the actual city road network, facilitating the accurate modeling of a risk propagating along the edges and intersections.

The proposed system—Bangalore Accident Routes—seeks to address limitations by combining a DBSCAN-based hotspots detection with a BallTree-based spatial influence modeling and a Bounded Multi-Source Shortest Path (BMSSP) algorithm to find the safest possible route between origin and destination points. The system uses real-time weather data from the OpenWeatherMap API to adaptively update its risk model, capturing environmental influences (including debris, rainfall, fog, or low visibility) that are

associated with an increase in accidents. This hybrid aspect has the potential to improve situational awareness and support adaptive routing under changing conditions.

A user-facing interactive web interface, designed utilizing Streamlit functionality, allows travelers to input their origin and destination parameters, configure clustering parameters, and visualize the safest route on a Folium-based map.

The application offers multiple route preferences—Shortest, Safest, and Balanced—thereby empowering users to make informed navigation decisions based on their personal safety preferences.

II. RELATED WORK

The examination of road traffic accidents and the preventive measures for accidents have gained an abundance of research attention as a result of the impact those incidents have on public safety and economic cost. There have been numerous proposals of interventions that target hotspots of danger, predictive frameworks, real-time detection, and routing for safer travel.

A substantial proportion of literature has implemented spatial analysis to identify hotspots for accidents. Kernel Density Estimation (KDE) has been widely implemented as a visualization tool for traffic accidents and accident hotspots; often, KDE fails to derive statistical validation and adaptation to be able to implement results in real-time. More recent works have employed clustering algorithms with the introduction of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and hierarchical agglomerative clustering to identify dense clusters of accidents, yielding a more valid approach for identifying and interpreting hotspots of accidents. One notable example by Hussien Kahm et al., 2024 utilized a model of DBSCAN for hotspots of accidents, specifying a GIS-based spatial analysis to obtain detailed mapping visualizing accident distributions. For the methods employed by the authors the methods came with computational costs, as well as, the inability to update dynamically with real-time variables.

Machine learning (ML) methods have now begun to slowly play a role for accident prediction and accident risk assessment. For example, methods such as Support Vector Machines (SVM), Random Forests, and ensemble learning methods have shown the validity of predicting accident severity or demonstrating the most contributing factors associated with historical indicators of accidents.
Notably, studies by Dhanya Viswanath et al. (2021) and Imad EL Mallahi et al. (2024) demonstrated improved prediction accuracy with ML models compared to traditional statistical methods, albeit with limitations due to dataset biases and limited real-time data integration.

Deep learning methods, in particular, Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNN), are being explored to provide evidence about the temporal and spatial patterns of accidents with respect to dynamic covariates and atmosphere conditions that include weather. Yulong Pei et al. (2024) introduced explainable AI methods to make a risk prediction model based in deep learning more interpretable, continued progression in the area toward relevance and usability.

Real-time accident monitoring applications combining AI and IoT methods—such as using computer vision-based applications to analyze video footage and using sensor-based networks—have been developed; however, they are challenged by costs, infrastructure development, and the reliability in conditions of poor visibility or limited connectivity.

Navigation and safety systems are mostly focused on determining a shortest or fastest route and do not have any complicated models for risk-aware routing using modelled accident data. This project is an attempt to synthesize a DBSCAN based hotspot identification, risk scoring of segments of road and potential safest route based on the Bounded Multi-Source Shortest Path (BMSSP) Algorithm; which is an iterative method integrating the results of spatial clustering, machine learning algorithms, and real-time data that contributes to improved risk-prevention and safe travel in a proactive manner.

III. PROPOSED WORK

The proposed system introduces a hybrid computational model that integrates spatial clustering, graph theory, and real-time environmental data to determine the safest travel routes within the Bengaluru metropolitan road network. The system leverages the power of OpenStreetMap (OSM) data, accident frequency analysis, and machine learning-based risk scoring to optimize navigation beyond conventional shortest-path computation.

A. System Architecture

The overall architecture of the proposed system is presented in Figure 1. It consists of four core modules interconnected in a data-driven pipeline:

1. Data Acquisition and Preprocessing Module
2. Accident Hotspot Detection Module
3. Risk Propagation and Weight Computation Module
4. Safe Route Computation and Visualization Module

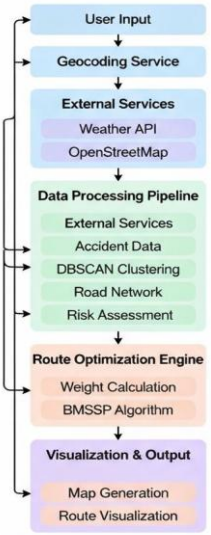

Figure 1: System Architecture Block Diagram

Each module is implemented using Python and geospatial data-processing libraries such as GeoPandas, OSMnx, Scikit-learn, and Folium, orchestrated within a Streamlit-based web interface for interactive exploration and visualization.

B. Data Acquisition and Preprocessing

1. Data Sources

Accident data for Bengaluru city are obtained from open government datasets and Kaggle repositories, comprising attributes such as:

1. Latitude and longitude of accident locations
2. Date and time of occurrence
3. Accident severity level
4. Weather condition at the time of incident

The spatial data are integrated with the Bengaluru street network retrieved using the OSMnx API, which models the city's transportation infrastructure as a weighted directed graph G= (V, E), where V represents intersections (nodes) and E denotes road segments (edges).

2. Preprocessing Steps

1. Geospatial Cleaning: Remove duplicate or incomplete entries and align all coordinate points to WGS84 (EPSG:4326).

2. Normalization: Standardize numerical fields such as severity and speed limit for clustering input.

3. Graph Construction: Convert OSM data into a routable NetworkX graph, embedding attributes like edge length, geometry, and travel direction.

C. Accident Hotspot Detection Using DBSCAN

To identify regions of frequent accidents, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is applied to geospatial coordinates. The clustering process groups accident locations into dense regions (hotspots) based on two parameters:

$$\varepsilon = \frac{r_m}{R_E}, \quad \text{and} \quad MinPts = k$$

where rm is the clustering radius in meters, RE=6,371,000 m is Earth's radius, and k is the minimum number of points required to form a cluster.

DBSCAN outputs clusters C={c1,c2,...,cn}, each representing a hotspot defined by its centroid, density, and accident frequency. The noise points are excluded from further processing.

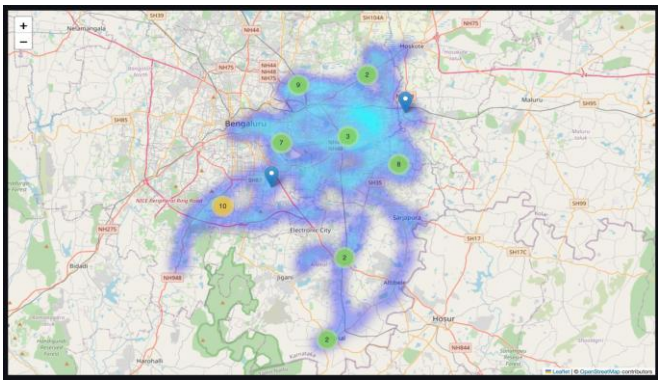

Figure 2: Map showing detected accident clusters in Bengaluru

The resulting centroids are stored in a spatial database for quick query access using KD-Tree and BallTree spatial indexing.

D. Risk Propagation Across the Road Network

To estimate the risk exposure of each road segment, the spatial influence of accident hotspots is propagated across the OSM graph using a distance-based exponential decay model.

For each edge $e_i \in E$, the geographic midpoint (lat_i, lon_i) is computed, and its distance d_{ij} to each hotspot h_j is determined using the Haversine distance:

$$d_{ij} = 2R_E \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_i)\cos(\phi_j)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

The base risk value for an edge is then calculated as:

$$R_i = \sum_{j=1}^{k} w_j e^{-\frac{d_{ij}}{\lambda}}$$

where

- w_j = normalized weight of hotspot h_j (based on accident count),
- λ = decay constant controlling influence range,
- k = number of nearest hotspots considered (typically 5).

E. Safety Weight Computation

Each edge is assigned a composite safety weight W_i that combines normalized distance and normalized risk:

$$W_i = \alpha \times \frac{L_i}{L_{max}} + \beta \times \frac{R_i'}{R_{max}}$$

where L_i is the edge length, L_{max} and R_{max} are maximum observed values for normalization, and α, β are user-defined parameters (with α+β = 1).

An additional penalty term is added for edges where Ri′>0.7, modeled as:

$$W_i^* = W_i + \delta, \quad \text{where } \delta = 50$$

This ensures avoidance of high-risk zones even if the distance advantage exists.

F. *Safe Route Computation Using BMSSP*

The routing algorithm is based on a modified Bounded Multi-Source Shortest Path Problem (BMSSP) formulation. Given the graph G(V,E), origin node s, destination node t, and an upper bound B on cumulative safety weight, the algorithm computes a path P* minimizing total weight:

$$P^* = \arg\min_{P \in \mathcal{P}(s,t)} \sum_{e_i \in P} W_i^* \quad \text{subject to} \quad \sum_{e_i \in P} W_i^* \leq B$$

The implementation combines Dijkstra's algorithm for shortest-path optimization and a breadth-first bound filter to limit search space.

The result is a path optimized for both minimal distance and minimal cumulative risk.

G. *Visualization and User Interaction*

The computed safe route is visualized using Folium maps, with the following layers:

- Hotspot Layer – Markers showing detected accident clusters.
- Risk Gradient Layer – Roads color-coded by risk intensity.
- Computed Route Layer – The safest path highlighted in green, showing travel distance and risk score.

The interactive visualization is integrated into a Streamlit web dashboard, which allows users to:

- Input origin and destination (with Nominatim geocoding).
- Adjust α and β coefficients to prioritize safety or distance.
- Set DBSCAN parameters (ε, MinPts) interactively.
- Choose route preference: Shortest, Safest, or Balanced.

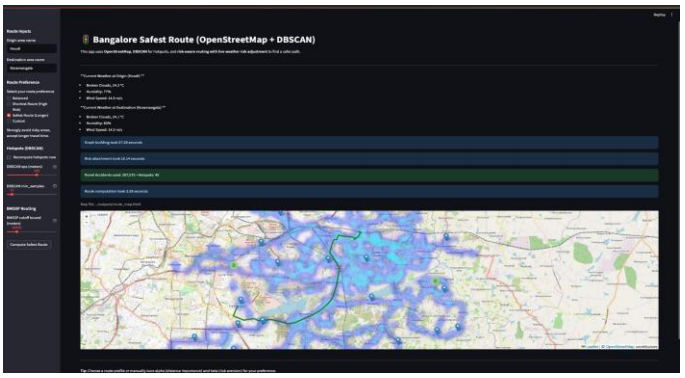

Figure 3: Screenshot of Streamlit interface showing input form and computed route on map

IV. ALGORITHMS

Algorithm 1: Finding Pivots

function FindPivots(B, S)

- **requirement:** foreveryincompletevertexvwithd(v) <B, the shortest path to *v* visits some complete vertex in S
- **returns:** sets P,W satisfying the conditions in Lemma 3.2

$W \leftarrow S$
$W_0 \leftarrow S$
for i ← 1 to k **do** ▷ Relax for k steps
  $W_i \leftarrow \emptyset$
  for each edge (u, v) with u ∈ W_{i-1} **do**
    if d[u] + w_{uv} ≤ d[v] **then**
      $d[v] \leftarrow d[u] + w_{uv}$
      if d[u] + w_{uv} < B **then**
        $W_i \leftarrow W_i \cup \{v\}$
  $W \leftarrow W \cup W_i$
  if |W| > k · |S| **then**
    $P \leftarrow S$
    return (P, W)
$F \leftarrow \{(u, v) \in E : u, v \in W, d[v] = d[u]+w_{uv}\}$
  ▷ F is a directed forest under Assumption 2.1
$P \leftarrow \{u \in S : u$ is a root of a tree with $\geq k$ vertices in $F\}$
return (P, W)

Algorithm 2: Base Case of BMSSP

function BaseCase(B, S)

- **requirement 1:** S= $\{x\}$is a singleton, and x is complete
- **requirement 2:** for every incomplete vertex v with $d(v) < B$, the shortest path to v visits x
- **returns 1:** a boundary $B' \leq B$
- **returns 2:** a set U

$U_0 \leftarrow S$
Initialize a binary heap H with a single element

⟨x, d[x]⟩
4: **while** H is non-empty and $|U_0| < k + 1$ **do**
5: ⟨u, d[u]⟩ ← H.ExtractMin()
6: $U_0 \leftarrow U_0 \cup \{u\}$
7: **for** each edge e = (u, v) do
8: **if** $d[u] + w_{uv} \leq d[v]$ and $d[u] + w_{uv} < B$ **then**
9: d[v] ← d[u] + w_{uv}
10: **if** v is not in H **then**
11: H.Insert(⟨v, d[v]⟩)
12: **else**
13: H.DecreaseKey(⟨v, d[v]⟩)
14: **if** $|U_0| \leq k$ **then**
15: **return** (B′ ← B, U ← U_0)
16: **else**
17: B′ ← max_{v ∈ U_0} d[v] 18:
U ← {v ∈ U_0 : d[v] < B′} 19:
return (B′, U)

Algorithm 3 Bounded Multi-Source Shortest Path

1: **function** BMSSP(l,B,S)
- **requirement1:** |S|≤2lt
- **requirement 2:** for every incomplete vertex x with d(x) < B, the shortest path to x visits some complete vertex y∈S
- **returns1:** aboundaryB′ ≤ B
- **returns 2:** a set U

2: **if** *l= 0* **then**
3: **return** B′,U ←BaseCase(B,S)
4: P,W ←FindPivots(B,S)
5: D.Initialize(M,B) with M = 2(l−1)t ▷ D is an instance of Lemma 3.3
6: D.Insert(⟨x,d[x]⟩) for x∈P
7: i←0;B′0 ←minx∈P d[x];U ←∅ ▷If P= ∅set B′0 ←B
8: **while** |U|<k2lt andD is non-empty **do**
9: i←i+ 1
10: Bi,Si ←D.Pull()
11: B′i,Ui ←BMSSP(l−1,Bi,Si)
12: U ←U ∪Ui
13: K ←∅
14: **for** edge e= (u,v) whereu∈Ui **do**
15: **if** d[u] + wuv ≤d[v] **then**
16: d[v] ←d[u] + wuv
17: **if** d[u] + wuv ∈[Bi,B) **then**
18: D.Insert(⟨v,d[u] + wuv⟩)
19: **else if** d[u] + wuv ∈[B′i,Bi) **then**
20: K ←K∪{⟨v,d[u] + wuv⟩}
21: D.BatchPrepend(K∪{⟨x,d[x]⟩: x∈Si andd[x] ∈[B′i,Bi)})
22: **return** B′←min{B′i,B};U ←U ∪{x∈W : d[x] <B′}

V. CONCLUSION

This study has successfully developed, executed, and assessed CrashGuard, an intelligent route optimization apparatus which implements safety as the primary calculable factor and pushes beyond traditional navigation. This project demonstrates a viable and realistic method for reduced travel risk in urban settings through the integration of historical crash data, spatial clustering, and risk-aware graph algorithms.

The central contribution to this work is its end-to-end pipeline that takes raw geospatial accident data and yields an interactive, user-friendly web application. The use of DBSCAN clustering enables the effective identification of accident hotspots without manually setting cluster counts, helping acknowledge the irregular and naturally forming shape of high-risk city pockets, such as Bangalore.

The next step involved establishing a risk-propagation model through the exponential decay function to visualize the aforementioned accident hotspots into a continuous risk field, modeling real world scenarios where a hotspot of danger would be capable of influencing other road segments in proximity.

A primary contribution of this work was the adaptation of the Bounded Multi-Source Shortest Path (BMSSP) algorithm. By creating a composite safety weight that balanced normalized distance against normalized risk, the system enabled adaptive routing capabilities, allowing the user to switch from routes that are "safest," to routes that are "shortest," and to as "balanced" routes as needed, depending on the functions of tunable parameters (α and β), thereby linking traditional navigation systems and safety based pathfinding systems.

Following the adaptation of the BMSSP algorithm, a Streamlit web dashboard was built as a demonstration of the system's ability to deliver real-world functionality. It was able to integrate geocoding, live weather data and interactive Folium maps into simple user experience and presenting users with data driven decisions for commuters and urban planners.

In total, the project demonstrates a tested, scalable framework for proactive management of road safety which shifts the focus from reactively analyzing accidents to proactively avoiding risk. Overall, the system represents a viable tool that could mitigate exposure to accidents and, thus, improve road safety and mobility in urban areas.

VI. REFERENCES

[1] K. Bringmann, A. Cassis, and N. Fischer, “Negative-weight single-source shortest paths in near-linear time: Now faster!,” in *Proc. IEEE 64th Annu. Symp. Found. Comput. Sci. (FOCS)*, Los Alamitos, CA, USA, Nov. 2023, pp. 515–538, IEEE Computer Society.
[2] A. Bernstein, D. Nanongkai, and C. Wulff-Nilsen, “Negative-weight single-source shortest paths in near-linear time,” in *Proc. IEEE 63rd Annu. Symp. Found. Comput. Sci. (FOCS)*, 2022, pp. 600–611.
[3] R. Duan, J. Mao, X. Shu, and L. Yin, “A randomized algorithm for single-source shortest path on undirected real-weighted graphs,” in *Proc. IEEE 64th Annu. Symp. Found. Comput. Sci. (FOCS)*, Los Alamitos, CA, USA, Nov. 2023, pp. 484–492, IEEE Computer Society.
[4] J. T. Fineman, “Single-source shortest paths with negative real weights in O(mn8/9)O(mn8/9) time,” in *Proc. 56th Annu. ACM Symp. Theory Comput. (STOC)*, New York, NY, USA, 2024, pp. 3–14, ACM.
[5] G. N. Frederickson, “Data structures for on-line updating of minimum spanning trees,” in *Proc. 15th Annu. ACM Symp. Theory Comput. (STOC)*, New York, NY, USA, 1983, pp. 252–257, ACM.
[6] M. L. Fredman and R. E. Tarjan, “Fibonacci heaps and their uses in improved network optimization algorithms,” *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987.
[7] T. Hagerup, “Improved shortest paths on the word RAM,” in *Proc. Int. Colloq. Automata, Languages, and Programming*, Berlin, Heidelberg: Springer, 2000, pp. 61–72.
[8] B. Haeupler, R. Hladík, V. Rozhoň, R. E. Tarjan, and J. Tětek, “Universal optimality of Dijkstra via beyond-worst-case heaps,” in *Proc. IEEE 65th Annu. Symp. Found. Comput. Sci. (FOCS)*, 2024.
[9] Y. Huang, P. Jin, and K. Quanrud, “Faster single-source shortest paths with negative real weights via proper hop distance,” 2024.
[10] D. R. Karger, P. N. Klein, and R. E. Tarjan, “A randomized linear-time algorithm to find minimum spanning trees,” *J. ACM*, vol. 42, no. 2, pp. 321–328, Mar. 1995.
[11] S. Pettie and V. Ramachandran, “A shortest path algorithm for real-weighted undirected graphs,” *SIAM J. Comput.*, vol. 34, no. 6, pp. 1398–1431, 2005.
[12] R. Raman, “Recent results on the single-source shortest paths problem,” *SIGACT News*, vol. 28, no. 2, pp. 81–87, Jun. 1997.
[13] M. Thorup, “Undirected single-source shortest paths with positive integer weights in linear time,” *J. ACM*, vol. 46, no. 3, pp. 362–394, May 1999.
[14] M. Thorup, “Floats, integers, and single-source shortest paths,” *J. Algorithms*, vol. 35, no. 2, pp. 189–201, May 2000.
[15] J. W. J. Williams, “Algorithm 232: Heapsort,” *Commun. ACM*, vol. 7, no. 6, pp. 347–348, 1964.
[16] V. Vassilevska Williams, “Nondecreasing paths in a weighted graph or: How to optimally read a train schedule,” *ACM Trans. Algorithms*, vol. 6, no. 4, Sep. 2010.
[17] M. Abdel-Aty and A. Radwan, “Modeling traffic accident occurrence and involvement,” *Accid. Anal. Prev.*, vol. 32, no. 5, pp. 633–642, 2000.
[18] A. Montella, “A comparative analysis of hotspot identification methods,” *Accid. Anal. Prev.*, vol. 42, no. 2, pp. 571–581, 2010.
[19] C. Xu, P. Liu, W. Wang, and Z. Li, “Identifying crash-prone traffic conditions using alternative data sources,” *Transp. Res. Part C: Emerg. Technol.*, vol. 38, pp. 511–524, 2014.
[20] C. Chen, G. Zhang, and H. Huang, “Developing a real-time accident risk prediction model using machine learning techniques,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1408–1417, 2016.
[21] M. Hossain and Y. Muromachi, “A Bayesian network-based framework for real-time crash prediction,” *Transp. Res. Rec.: J. Transp. Res. Board*, no. 2279, pp. 19–28, 2012.
[22] S. Washington, M. Karlaftis, and F. Mannering, *Statistical and Econometric Methods for Transportation Data Analysis*, 2nd ed., Boca Raton, FL, USA: Chapman and Hall/CRC Press, 2010.
[23] S. Das and X. Sun, “A machine learning approach to accident prediction using roadway characteristics and environmental factors,” *J. Saf. Res.*, vol. 58, pp. 79–87, 2016.
[24] K. M. Kockelman and Y. J. Kweon, “Driver injury severity: An application of ordered probit models,” *Accid. Anal. Prev.*, vol. 34, no. 3, pp. 313–321, 2002.
[25] S. Singh, “Critical reasons for crashes investigated in the national motor vehicle crash causation survey,” *Nat. Highway Traffic Safety Admin. (NHTSA)*, Rep. No. DOT HS 812 115, 2015.
[26] P. S. Lin and H. Yu, “Use of AI in traffic accident prediction and risk management,” *J. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 55–67, 2019.