Real-Time Driver and Traffic Data Integration for Enhanced Road Safety

Yufei Huang, Shan Jiang, Mohsen Jafari, and Peter J. Jin

Abstract—Traditional roadway safety assessment heavily relies on historical crash data, overlooking real-time factors such as driver behaviors and current traffic conditions and lacking forward-looking analysis for predicting future trends. This study introduces an enhanced innovative data fusion method based on the safe route mapping (SRM) methodology with combined use of historical crash data and real-time data, leveraging a custom-built Android app to amalgamate road and vehicle data effectively, showcasing notable advancements in real-time risk assessment. The enhanced safe route mapping (ESRM) framework monitors driver actions and road conditions meticulously. Data collected from drivers is analyzed on a central server using facial recognition algorithm to detect signs of fatigue and distractions, assessing overall driving competence. Simultaneously, roadside cameras capture live traffic data, analyzed using a specialized video analytics method to track vehicle speed and paths. The fusion of these data streams enables the introduction of a predictive model, Light gradient boosting machine (GBM), forecasting potential immediate issues for drivers. Predicted risk scores are integrated with historical crash data using a Fuzzy logic model, delineating risk levels for different road sections. The performance of ESRM model is tested using real-world data and a driving simulation, demonstrating remarkable accuracy, especially in accounting for real-time fusion of driver behavior and traffic conditions. The resultant visual risk heatmap aids authorities in identifying safer routes, proactive law enforcement deployment, and informed trip planning based on real-time risk levels. This study not only underscores the importance of real-time data in roadway safety but also paves the way for data-driven, dynamic risk assessment models, potentially reducing road accidents and fostering a safer driving environment.

Index Terms—Accident prevention technology, driver behavior monitoring, predictive safety modeling, real-time traffic analysis, road safety enhancement.

I. INTRODUCTION

MOTOR vehicle crashes pose a significant societal and economic threat worldwide, necessitating effective measures for roadway safety assessment and management. Traditional approaches have relied heavily on historical crash data and qualitative measures to identify hazardous road features and devise countermeasures. However, these methods often fall short in addressing rapidly evolving factors such as traffic congestion, adverse weather conditions, and driver behavior, which are crucial in real-time risk assessment and management [1]. Additionally, the lack of precrash data makes it difficult to study conflict probability, predict conflicts, and provide advance warnings [2]. Moreover, human errors, including speeding, fatigue, drunk driving, and distracted driving, account for over 90% of vehicle crashes [3]. Yet, widely used trajectory datasets for analyzing traffic conflicts, such as the HighD dataset [4] and next-generation simulation (NGSIM) vehicle trajectories and supporting data [5], often do not consider drivers’ behavior during accidents due to the absence of in-vehicle information. Advancements in communication and vehicle-to-infrastructure (V2I) technologies present an opportunity to harness dynamic traffic conflict data and vehicle interior information [6]. Through the use of roadside infrastructures, such as cameras and LiDAR, efficient vehicle detection and tracking can generate speed and trajectory profiles [7]. In-vehicle cameras can also capture driver behaviors, which can be uploaded to roadside servers for analysis. By combining roadside information with driver behaviors, real-time prediction of traffic conflicts becomes possible, providing early warnings to individual road users. Furthermore, the assessment of overall roadway risks can assist transportation authorities in enforcing safety regulations, utilizing the insights gained to implement targeted interventions and improve overall road safety.

In road safety studies, three common methods are used to evaluate how effective safety changes are. Safety performance functions (SPFs) [8] predict the expected number of crashes at a place based on traffic and road features. The empirical Bayes (EB) method [9] improves these predictions by adjusting them with actual crash data, which helps to correct any biases. Crash modification factors (CMFs) [10] measure how much a specific safety feature can reduce crashes based on data from before and after the feature was implemented. Unlike these traditional methods, our study uses a cross-sectional approach. This means we look at crash data from different locations at one point in time rather than looking at changes over time at the same
location. This approach helps us to understand which factors are currently affecting road safety across various sites.

This research, in its pursuit to address the critical disconnect between theoretical traffic safety advancements and their practical application, builds upon our initial efforts with the safe route mapping (SRM). The primary challenge identified has been the difficulty in acquiring real-time driver behavior data, often restricted due to the need for additional in-vehicle equipment. This limitation impacts the accuracy of driving performance assessment and highlights deficiencies in existing roadway safety prediction models, particularly in their failure to consider crucial individual driver attributes, such as emotional states and attention levels. Consequently, there is a pressing need for a more comprehensive and integrated model, one that can encapsulate a wide array of inputs for a more precise and holistic understanding of driving safety. Such a model would address the current neglect of the complex interplay of human factors in driving safety assessments. Building on this premise, we developed the enhanced safe route mapping (ESRM) as a significant advancement in the field. The ESRM not only integrates a broader spectrum of data but also focuses specifically on those aspects related to human factors. This development marks a substantial stride in enriching our understanding of driving safety, offering considerable improvements in predictive accuracy and prevention strategies. It represents a critical step in merging theoretical knowledge with practical application within the domain of traffic safety analysis. The current study furthers the work on the SRM methodology, as initially developed and detailed in earlier research [11], [12], to enhance the calculation of roadway risk levels. The original SRM model made commendable progress by integrating crash-based estimates with conflict risks computed from driver-based data to evaluate roadway safety. It used an advanced SPF and a driver-based model for dynamic risk assessment from driver and traffic data. However, the reliance on simulated driver data in the SRM posed challenges in accurately reflecting real-world driving conditions. To address these limitations, the present study enhances the SRM model by incorporating real-world data collected through a developed Android application for drivers and roadside cameras for live traffic data. This fusion of real-time driver behavior with live traffic data, when integrated into the SRM model, facilitates a nuanced understanding of roadway risks, surpassing the original methodology’s reliance on simulated data. This enhanced SRM model leverages real-world driver behavior data and live traffic data, leading to more accurate predictions of roadway safety scores and considering human factors previously underrepresented.

The real-time data collection and analysis not only offer a better understanding of current roadway conditions but also enable immediate response measures, such as dispatching law enforcement and alerting drivers to potential hazards. This study contributes in three significant ways: it proposes an innovative method to simultaneously collect traffic data from both the roadside and inside vehicles, presents an improved SRM model, that is, the ESRM, incorporating human factors for enhanced prediction accuracy, and generates risk heat maps that assist authorities in identifying safe corridors and provide drivers with timely alerts about high-risk areas.

This article is structured as follows. Section II reviews measurement methods and computer vision techniques essential for recognizing vehicle trajectories and evaluating driving performance. Section III illustrates the extended SRM algorithm, including the application of the SPF model, a light gradient boosting machine (LightGBM), and a fuzzy logic model for risk score calculation. Section IV describes the data collection process and the method employed to obtain vehicle trajectory data. Section V presents the model training process and validation results of our proposed method. Section VI introduces a simulation platform for additional data gathering and validation of the improved SRM model. Finally, Section VII provides a summary and discusses future considerations.

II. LITERATURE REVIEW

The Highway Safety Manual (HSM) published by the American Association of State Highway and Transportation Officials (AASHTO) recommends the use of SPFs to make informed safety decisions [13]. However, the current practice of using SPFs to predict average crash frequency often suffers from limited data and model accuracy [14]. Since vehicle crashes are rare events compared to the large volume of daily traffic, incorporating extreme value theory and studying traffic conflicts or near misses can provide a rough approximation for understanding traffic accidents, enhanced by extreme value theory’s focus on statistical extremes [15]. Near misses, which refer to accidents that were narrowly avoided, can be identified using specific traffic conflict indicators (TCIs) [16]. Considering that human error accounts for approximately 94% of all vehicle crashes, naturalistic driving studies (NDSs) have been conducted to observe and record driver behavior in real time [17]. Driver-based data from NDS have been used to analyze collisions, develop crash surrogates, and create collision avoidance advisory systems [18]. One example is the 100-car NDS, sponsored by the National Highway Traffic Safety Administration (NHTSA) and Virginia Department of Transportation (VDOT), which collected large-scale naturalistic driving data [19]. However, both driver- and vehicle-based data approaches still face challenges such as data reliability and low data density [11].

In recent years, with the advancement of V2I technology and the decreasing cost of infrastructure sensing, it has become possible to collect data from inside vehicles and from roadside infrastructure. Datasets such as the HighD dataset, recorded on German highways using a drone, and the NGSIM dataset, which includes vehicle trajectories, have been widely used for studying traffic behavior. However, these datasets may not fully address the complexities of traffic situations in urban areas with intersections.

Computer vision sensors integrated into traffic surveillance systems, along with cloud computing, offer new opportunities for enhancing traffic safety. The main steps of traffic video analysis using these sensors are vehicle detection and tracking. The detection step involves separating vehicles from the background, while the tracking step traces their movements. Popular model-based methods for localizing and classifying vehicles based on their shapes include faster region proposal convolutional

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
neural network (faster R-CNN) [20], single shot MultiBox detector (SSD) [21], and You Only Look Once (YOLO) [22]. In a previous study [23], a computer vision algorithm was introduced for obtaining vehicle trajectories from high-angle traffic videos. This algorithm combines scanline-based trajectory extraction and feature-matching coordinate transformation to detect vehicles and track their movements. The method has been proven to be robust and accurate in obtaining vehicle traces from the video. Inside a vehicle, a driver concentration control system can monitor the driver’s behavior using various methods such as cameras, eye trackers, and contactless sensors. One solution proposed in [24] involves using a smartphone application, which is more affordable compared to adding additional devices inside the vehicle. Videos of the driver are streamed to a server, where a facial recognition system can analyze the driver’s performance.

Deep learning techniques, such as convolutional neural networks (CNNs), have gained popularity in face detection, face alignment, and feature extraction tasks. Facial behavior analysis toolkits such as OpenFace can monitor facial landmark motion, head pose (orientation and motion), and eye gaze [25]. Studies have shown that the driver’s emotions can affect driving performance, and a U-shaped relationship between performance, arousal, and valence can be modeled [26]. Facial expressions can be identified using the facial action coding system (FACS) based on the facial landmark motion [27]. Additionally, a drowsiness alarm can be designed based on the eye aspect ratio (EAR) calculated from facial landmarks to detect driver drowsiness [28]. Head pose and eye gaze can also indicate whether the driver is focusing on the road, which is crucial for monitoring driver concentration.

The analysis of NDSs has shown correlations between driver behaviors and roadway segments in both crash/near miss and regular driving situations [29]. In [29], a data-driven approach is proposed for drivers’ safety risk profiling and roadway segments’ safety risk scoring. Elastic net regularized multinomial logistic regression is applied to enhance the prediction performance of the road safety model. In [11], the model is further developed using neural networks, and risk heat maps are created for visualization purposes. However, the driver behaviors considered in [11] are limited to actions such as speeding, stop sign violations, and rapid lane changes, which may not be comprehensive enough to capture all relevant driver behaviors related to safety risks on the road. Further research may be needed to incorporate a wider range of driver behaviors into the safety risk profiling model to improve its accuracy and effectiveness in identifying safety risks on roadway segments.

III. IMPROVED SRM MODEL

This section presents a methodology for dynamic risk heat mapping of road segments using multiple-sourced data. First, a crash prediction model is introduced to estimate the expected number of crashes. Next, a LightGBM model is employed to predict TCIs for individual drivers. Finally, a fuzzy logic model is applied to combine the expected number of crashes and TCIs, resulting in risk scores and risk heat maps for road segments. The entire workflow is presented in Fig. 1.

A. Expected Number of Crashes

In this section, we present a SPF model using real crash data formulated with a negative binomial (NB) distribution to predict crash counts more accurately. This model addresses the over-dispersion commonly found in crash data, where variance surpasses the mean, making traditional Poisson models inadequate due to their equal mean and variance assumption. We utilize the EB [30] method to refine our estimates. Generally, crash occurrences \( N_i(t) \) at site \( i \) during year \( t \) are assumed to follow a Poisson distribution:

\[
f(N_i(t), \lambda_i) = e^{-\lambda_i} \frac{(\lambda_i)^{N_i}}{N_i!}
\]

where \( \lambda_i \) is the rate parameter. However, since crash data often exhibit over-dispersion, we adopt the NB distribution by integrating the Poisson model with a Gamma distribution for \( \lambda_i \), allowing for variance that is greater than the mean:

\[
f(N_i(t)|\lambda_i, \nu, \delta) = \int_0^\infty e^{-\lambda_i} \left( \frac{\lambda_i}{\nu} \right)^{N_i} \frac{\Gamma(\nu + N_i)}{\Gamma(\nu) \Gamma(N_i + 1)} \left( \frac{\delta}{1 + \delta} \right)^\nu \left( \frac{1}{1 + \delta} \right)^{N_i} d\lambda_i.
\]

Here, \( G(\lambda|\nu, \delta) \) denotes the Gamma distribution with shape \( \nu \) and rate \( \delta \), which adjusts \( \lambda_i \) dynamically. By leveraging the properties of the Gamma distribution, we can simplify the integration, resulting in

\[
f(N_i|\nu, \delta) = \frac{\Gamma(\nu + N_i)}{\Gamma(\nu) \Gamma(N_i + 1)} \left( \frac{\delta}{1 + \delta} \right)^\nu \left( \frac{1}{1 + \delta} \right)^{N_i}.
\]

This equation bears a resemblance to the type 2 negative binomial (NB2) distribution, with parameters \( r = \nu \) and \( p = (1 + \delta)^{-1} \). The mean of the NB2 distribution can be expressed as \( E(N_i) = \mu_i = \nu / \delta \), and its variance as \( \text{Var}(N_i) = \mu_i + \mu_i^2 / \nu \). This indicates that the variance of crash counts is typically higher than the mean, making NB2 suitable for handling over-dispersion. For sites with uniform features, the expected number of crashes \( E(N_i) \) is modeled as

\[
E(N_i) = \mu_i = e^{x_i^T \beta_i},
\]

where \( x_j \) represents roadway features, as listed in [11], and \( \beta_i \)'s are estimated using the maximum likelihood estimator (MLE). We apply these coefficients within a logarithmic link function that reflects the influence of each roadway feature.
on the crash probabilities
\[
\ell(N_i|x, \beta, \alpha) = \sum_{i=1}^{n} \left( N_i \ln \frac{\alpha \exp(x_i/\beta)}{1 + \alpha \exp(x_i/\beta)} - \ln(1 + \alpha \exp(x_i/\beta)) \right) + C_i
\]  

(5)

In the equation, \( \alpha = 1/\nu \) and \( C_i \) adjusts for the gamma function normalization. Further refining our model, we use the EB method, where the posterior distribution of \( \lambda_i \) follows a gamma distribution, allowing us to estimate \( \lambda_i \) based on historical crash data
\[
E(\lambda_i|N_i) = \frac{\nu + N_i}{1 + \delta}
\]

(6)

Recall \( \alpha = 1/\nu \) and \( \delta = 1/\nu \mu_i \), we can further express (6) as
\[
E(\lambda_i|N_i) = \left( \frac{1}{1 + \delta \mu_i} \right) \mu_i \left( 1 - \frac{1}{1 + \delta \mu_i} \right) N_i
\]

(7)

This formulation ensures that the expected number of crashes at each site is a weighted average of the model’s predictions and the observed data, enhancing the reliability of our crash frequency estimates.

B. Traffic Conflict Prediction

TCIs are adopted to study near misses, which occur more frequently than collisions. Analyzing TCIs can help notify drivers to take action before collisions happen. However, research has shown that adopting different indicators will give us different labels of conflicts [31]. Therefore, we combine the results provided by various indicators in this study. We use three commonly used TCIs that are as follows: 1) time to collision (TTC); 2) modified time to collision (MTTC); and 3) deceleration rate to avoid a crash (DRAC). TTC \( (T_t) \) estimates the time to a crash based on current distance and speed and is defined in (8). \( x_L \) and \( v_L \) represent the longitudinal coordinate and speed of the leading vehicle. \( x_F \) and \( v_F \) represent the longitudinal coordinate and speed of the following vehicle. The commonly used threshold value for TTC is 1.5 s. TTC smaller than 1.5 s implies a near miss. MTTC \( (T_m) \) improves TTC by considering acceleration and is defined in (9). \( \Delta v \) means the speed difference between the leading and following vehicles. \( \Delta a \) means the acceleration difference between the two cars. The commonly used threshold for MTTC is also 1.5 s. A near miss happens when MTTC is less than 1.5 s. Finally, DRAC \( (I_d) \) shows how much deceleration the following vehicle needs to avoid crashing into the front car. It is defined in (10). The threshold value for DRAC is 3.35 m/s². It is safe when DRAC is smaller than the threshold
\[
I_T = \frac{x_L - x_F}{v_L - v_F}
\]

(8)

\[
I_M = \frac{\Delta v \pm \sqrt{\Delta v^2 + 2\Delta a(x_L - x_F)}}{\Delta a}
\]

(9)

\[
I_D = \frac{(v_F - v_L)^2}{2(x_L - x_F)}
\]

(10)

A LGBM [32] was introduced to predict whether TTC, MTTC, or DRAC will go beyond the threshold in the next 1 or 2 s. LGBM is a gradient-boosting framework that uses tree-based learning algorithms. Instead of building an ensemble of deep independent trees (random forest), LGBM creates an ensemble of shallow and weak successive trees. Each tree learns and improves the previous results. LGBM has faster training speed and accuracy than any other boosting algorithm. Fig. 2 illustrates the LGBM structure used in this work. The inputs can be classified into six categories.

1) “Driver behavior” includes the features we create in Section III-C.

2) “Road characteristic” refers to the geometric configuration of the lane in which the vehicle is running. This variable is represented by dummy variables that categorize the lane based on its directional changes, specifically indicating whether it diverges, merges, turns left or right. Each category is defined by specific criteria related to the lane’s layout and the driver’s maneuvering options at given points on the road.

3) “Vehicle status” shows whether the vehicle is leading or following another car. “In queue” is a binary variable. Direction means the yaw angle of the car. The angle to the front vehicle shows the angle between our vehicle’s heading direction and the direction to the front vehicle.

4) “Distance” includes headway distance (HWDY), which is the distance between two successive vehicles’ front bumpers.

5) “Speed” contains the vehicle’s current speed and acceleration. It also indicates the speed difference between the front and host vehicles.

6) The last category contains TTC, MTTC, or DRAC values in the previous 1 s and the current time. For example, when we want to predict TTC in the next 1 or 2 s, we put the TTC value in the previous one and at the
current time point here. When predicting MTTC or DRAC, we replace these inputs with the corresponding values. Therefore, we use six LGBM models with a similar structure for predicting these indicators.

The GPS coordinates, speed, and acceleration data can be obtained from both roadside video and the Android APP. The duplicated information is combined to increase the system’s reliability. The values collected from two data sources are averaged to reduce observational errors. When one data source is unavailable due to network latency from the phone or miss-recognition from the video, the other data source will be applied.

The output of each LGBM model is a binary variable that indicates whether one of the three TCIs will go beyond the threshold in the next 1 or 2 s. The outputs can take values of either 0 (no conflict) or 1 (conflict).

Note that not all vehicles have in-vehicle data. For the drivers who cannot or are not willing to use the Android APP to share their information, a simplified LGBM model is trained and used. The simplified version removes the inputs in the category “driver behavior.”

C. Risk Score

Fig. 3 provides an example of how road segments are divided based on the direction of the lane at an intersection. The intersection is divided into seven parts, each corresponding to a different direction of the lane. These road segments are used to calculate the risk scores for each vehicle based on the predicted TCIs and the historical crash data using the fuzzy logic model, as described in the previous response. The risk scores provide an indication of the level of risk associated with each road segment for each vehicle at each time point, helping to identify potential conflict areas and improve road safety measures.

The fuzzy logic model used in this work combines the expected number of crashes per year of the road segment with the prediction of three TCIs (TTC, MTTC, and DRAC) at each time point. The expected number of crashes per year is categorized into three levels: low, medium, and high. These levels are defined by the thresholds of 0–5, 6–10, and over ten crashes per year, respectively. These thresholds were initially chosen based on a preliminary analysis of the distribution of crash data within our dataset, which suggested these intervals as meaningful dividers that reflect distinct levels of traffic safety risk. To validate and refine these thresholds, further statistical analyses and expert consultations are planned, ensuring they accurately represent varying degrees of safety conditions and align with standard practices in traffic risk assessment. The fuzzy logic model calculates the risk score on a scale of 0–100 using 24 rules, as described in Table I. The risk score of a road segment is determined by aggregating the individual risk scores of all vehicles present in that segment at each specific time point. Specifically, this aggregation is performed by identifying the maximum risk score recorded among all vehicles in the segment during the same time instant. This approach is chosen because it reflects the worst-case scenario at any given moment, which is critical for assessing and mitigating potential hazards effectively.

Fig. 4 illustrates the structure of the fuzzy logic model, which takes into account the historical crash data and real-time TCI predictions. The TCI predictions can take values of either 0 (indicating no conflict) or 1 (indicating conflict). The fuzzy logic model uses these TCI predictions along with the expected number of crashes per year to calculate the risk score for each road segment. This risk score provides an indication of the level of risk associated with that road segment based on historical crash data and real-time TCI predictions.

D. Risk Heat Map

The risk heat map is created to visualize the risk scores for each segment. The risk scores are divided into five levels: very small [if the risk score falls in [0, 20]], small [if the risk score falls in [20, 40]], medium [if the risk score falls in [40, 60]], large [if the risk score falls in [60, 80]], and very large [if the risk score falls in [80, 100]]. The corresponding colors assigned to each level are green, blue, yellow, orange, and red. Based on these rules, a cell plot is created to show the value of risk scores and the predicted risk scores for each segment at each time point. Fig. 5 shows an example of the risk heat map. We anticipate the risk scores for each road segment at every time point. Then, the actual risk scores are calculated based on the
factual data collected in the next 1 or 2 s to validate the prediction results. In the plot, each row corresponds to a road segment in Fig. 3. The first and third columns show the actual risk score in the next 1 or 2 s, and the second and fourth columns are our predicted results for the risk score in the next 1 or 2 s. By taking an average of the risk scores of each road segment in specific time windows, Fig. 6 illustrates the changes in road risks through time. Also, Google maps is customized to show the prediction results using Google map API, as shown in Fig. 7. The personalized map can provide a clear view of the safety situation on the road, and the map will be sent back to the Android APP. The APP users can be alerted about the predicted high-risk area during a trip.

IV. DATA COLLECTION

In this section, the data collection and preprocessing steps shall be discussed. The section introduced the architecture of the data collection process that retrieves the videos from the roadside infrastructure and inside the vehicles. Also, the applied vehicle trajectory recognition and drivers’ behavior detection methods are presented.

A. Data Flow Diagram

Fig. 8 shows the data flow diagram of the data collection process. First, raw data collected by the roadside camera and Android APP are uploaded to the cloud server via the 5G network. The vehicle profile created by vehicle trajectory recognition and the driver behavior data created by the driver’s face recognition are synchronized and merged based on the GPS information (longitude and latitude) and timestamp. For example, suppose the Android APP reports the vehicle at specific GPS coordinates that are also detected by the roadside camera simultaneously. In that case, we then conclude they represent the exact vehicle. Next, python programs generate vehicle trajectory and driver behavior profiles using the uploaded raw data in real time. Finally, risk scores for each road segment are predicted using the ESRM. The risk heat maps are created based on the risks scores and are transmitted to the Android APP to warn drivers about the traffic conflict probabilities. The ESRM and the risk heat map will be discussed in Section IV. AWS is selected to host the database and perform cloud computing in this work.

B. Vehicle Trajectory Recognition

Videos of the roadway are captured by a camera positioned on the roadside infrastructure and transmitted to a cloud server in real time. On the server, a scanline-based trajectory extraction technique, originally proposed in [23], is employed to identify vehicles and extract their trajectories from the videos. This technique utilizes color, gradient, and motion features [33] to separate vehicle strands from the pavement background on a spatial–temporal diagram, as depicted in Fig. 9. The actual location of a point in the video can be estimated using coordinate transformation with Google maps. The front bumper of each vehicle is selected as the representative position on each frame. By utilizing the video recorded at a frame rate of 20 frames per second and the real-world coordinates, speed, and acceleration information for each vehicle at each timestamp can be...
calculated. During the data preprocessing phase, additional features are generated based on the raw data from the roadside camera, which can be categorized into four types: 1) roadway characteristics—such as lane shape, including going straight, turning left, turning right, merging, and diverging; 2) vehicle status—including whether the vehicle is leading a queue or following other vehicles, and the angle between its direction and the front car; 3) speed profile—including speed, acceleration, and speed difference to the front car; and 4) distance profile—including headway and safety distance based on vehicle type\[11\]. These features will serve as input variables in our traffic conflict prediction model, which was discussed previously in Section III.

C. Driver Behavior Analysis

An Android application is developed to collect drivers’ behavior data inside a vehicle. Drivers can place their Android phones on the dashboard or an air vent in a phone holder, with the front camera aimed at the driver’s face, as shown in Fig. 10.

The application is designed to capture video of the driver’s face, which is then uploaded to the cloud server. In addition, vehicle speed, acceleration, and GPS information are also recorded and uploaded to the cloud server along with the video. The driver’s video is processed on the server using the OpenFace toolkit, which is an open-source facial behavior analysis toolkit capable of detecting facial landmarks, estimating head pose and eye gaze, and recognizing facial action units (AUs) [34]. OpenFace utilizes deep neural networks to represent the face on a 128-dimensional unit hypersphere. It can detect 68 facial landmarks, as shown in Fig. 10(a).

These landmarks are critical points on a human face that localize and label the facial regions such as the mouth, eyebrows, eyes, nose, and jaw. OpenFace utilizes deep neural networks to represent the face on a 128-dimensional unit hypersphere. It can detect 68 facial landmarks, as shown in Fig. 11(a).

These landmarks are critical points on a human face that localize and label the facial regions such as mouths, eyebrows, eyes, noses, and jaws. In addition to estimating facial landmarks, OpenFace also provides information about the angles of head pose and eye gaze, as shown in Fig. 11(b). Assume \( x_h \) and \( y_h \) as the angle of head pose on x- and y-axis. Let \( x_e \) and \( y_e \) be the angle of the eye gaze on x- and y-axis. Using (11), we can calculate the angle of the driver’s focusing area represented by the angle on the x-axis \( x_f \) and the angle on the y-axis \( y_f \). \( x_f \) is positive when the driver is looking to the left, and it is negative when the driver is looking to the right. \( y_f \) is positive when the
The FACS is used to determine the AUs based on their descriptions. Each emotion is characterized by specific and measurable facial muscle movements, which can be quantified using a combination of AUs. For example, happiness is determined by the joint activities of facial muscles associated with the inner brow raiser, outer brow raiser, cheek raiser, and nose wrinkles. Facial landmarks around the eyes are considered closed when the eyes are closed, as shown in Fig. 12. The eyes are considered open when the pupils are dilated, as shown in Fig. 13. The coordinates of each emotion on the valence-arousal (VA) plane can be found using the following equations:

\[
\begin{align*}
    x_f &= x_b + x_c \\
    y_f &= y_b + y_c
\end{align*}
\]

(11)

K-means clustering [35] is applied to congregate the points that represent where the driver is looking using the focusing area coordinates \((x_f, y_f)\). As shown in Fig. 12, the precollected focusing points are grouped into three clusters. Based on where each point is locating at, we can approximate whether the driver is looking straight ahead or not.

EAR is commonly used for fatigue level classification. If the eye-blinking frequency radically increases, we will observe continuous changes in EAR [36]. EAR \((r_e)\) can be calculated using (12) given the coordinates of the facial landmarks around the eyes, as shown in Fig. 13:

\[
r_e = \frac{\sqrt{(x_2 - x_b)^2 + (y_2 - y_b)^2 + (x_3 - x_s)^2 + (y_3 - y_s)^2}}{2 \cdot \sqrt{(x_4 - x_1)^2 + (y_4 - y_1)^2}}.
\]

(12)

The threshold of \(r_e\) is set at 0.26 based on an experiment shown in Fig. 14. The eyes are considered closed when \(r_e\) is smaller than the threshold.

Similarly, the mouth aspect ratio (MAR) is used to indicate whether the driver’s mouth is open or not. The threshold of MAR \((r_m)\) is set as 0.05 based on experiments. If \(r_m\) goes beyond the threshold, and the driver might be talking or eating, which distracts the driver from focusing on the driving task.

In addition to the driver’s focus and fatigue level, emotions also have an impact on human performance by influencing judgment and behavior. For instance, stressed operators may struggle to achieve optimal performance in complex task environments [37]. The FACS is used to determine a driver’s emotion, which involves identifying specific facial muscle movements associated with displayed emotions. FACS consists of AUs [27], which encompass 46 main AUs such as inner brow raiser, outer brow raiser, cheek raiser, and nose wrinkle; eight head movement AUs including head turn left, head turn right, head up, and head down; and four eye movement AUs including eyes turn left, right, up, and down. Each emotion is characterized by specific joint activities of facial muscles. For example, happiness is determined by the combination of AUs 6 (cheek raiser) and 12 (lip corner puller), while sadness is identified by unit 1 (inner brow raiser), unit 4 (brow lowered), and unit 15 (lip corner depressor). Facial landmarks movement is used to determine the AUs based on their definition.

In this work, we focus on six basic emotions: happiness, sadness, surprise, fear, anger, and calm. Cai and Lin [26] proposed a way to quantify emotions using a valence–arousal (VA) plane, as shown in Fig. 15. This 2-D emotion model allows for the representation of emotion combinations rather than pure emotions. The coordinates of each emotion on the VA plane can be found in Table II. According to the Yerkes–Dodson law, performance tends to increase with physiological or mental arousal up to a certain point, after which it decreases [38]. This relationship is often depicted as a bell-shaped curve, as shown in Fig. 16. A bivariate normal distribution can be used to model the relationship between emotion and performance in this 2-D space. Experiments that utilize various measures to assess driving
performance have shown that different optimal performance points, also known as “sweet points,” can be achieved. Based on [26], when driving performance is defined based on the number of traffic violations, the sweet point coordinates on the VA plane are (0,0.2). When lane deviation is used as the performance judgment, the sweet point is located at (0,0.1). Similarly, if brake reaction time is considered as the performance measure, the sweet point coordinates are (0.2,0.2). Notably, all three sweet points fall within the calm area of the VA plane, indicating that optimal driving performance is achieved when the driver is calm. To calculate the driver’s performance score, the probability density functions (PDFs) of the three bivariate normally distributed models are summed. Each model corresponds to one of the three sweet points identified earlier.

To conclude, we have introduced four additional features derived from the driver’s behavior, namely focusing area, EAR, MAR, and performance score based on emotion. These features are utilized as inputs in the traffic conflict prediction model, which was discussed earlier in Section III.

V. MODEL TRAINING AND VALIDATION

This section demonstrates how the SRM model is trained using actual traffic data collected from an intersection. The validation results are also presented.

A. Scope of the Testbed

The testbed is located at the intersection of Easton Avenue and Albany Street in New Brunswick, NJ, USA. The roadside camera is placed on the roof of a public garage. It provides a bird’s eye view of the intersection, as shown in Fig. 9. The New Brunswick train station located at the intersection makes the traffic conditions more complicated.

Four test drivers were invited to join the experiment and used the Android APP to collect their facial video during the test, as shown in Fig. 17. The experiment was performed at 3:35 pm on 16 May 2021, and it lasted 40 min. The data collected from the experiment was divided into two parts. In the first 20 min, the collected roadside video and drivers’ information are used to train the LGBM model mentioned in Section III-B. In the second half, we evaluate the accuracy of the LGBM model and the fuzzy logic reasoning model using the rest of the data. There are 1833 recognized vehicle trajectories, including the exact vehicle with multiple trips passing through the area. The test drivers were asked to keep driving around and driving through the testbed as often as possible.

Among all the recognized trajectories, 36 trajectories were made by the four test drivers. Their driving behavior information was matched with their speed profiles collected by roadside video. For the other trajectories that miss the driver’s information, a simplified LGBM model, which has no driver behavior inputs, was trained separately, as discussed in Section III-B.

B. Validation Results

In our study, we utilized the 100-driver dataset [39] to validate our facial recognition algorithms for detecting driver fatigue and distraction as discussed in Section IV-C. This dataset is particularly suited for our purposes as it provides video data with labeled driver behaviors including sleeping, looking down, looking right, looking left, and driving safely across multiple camera views (front, left, and right). OpenFace is applied on these labeled videos, and the proposed fatigue and distraction recognition algorithm is tested. The results demonstrate the effectiveness of our model, with accuracies ranging from 72.1%
for “normal driving,” 67.3% for “sleeping,” 91.7% for “looking left,” and 94.9% for “looking right.”

One limitation of OpenFace is that the facial recognition result is not stable in poor illumination conditions. The driver’s face can be detected in low-light conditions, as shown in Fig. 18(a). The testbed has enough streetlights at night, and the model works. However, the driver’s face cannot be seen when it is too dark, as shown in Fig. 18(b). When streetlights are not bright enough, the authors suggest removing driver behavior features from the proposed LGBM model and using the simplified model to predict the safety index in dark environments.

Notice that 16.2% of the training data detect a conflict based on either TTC, MTTC, or DRAC, indicating that the classification’s training set is imbalanced. The synthetic minority oversampling technique (SMOTE) is utilized to deal with the problem. SMOTE handles an imbalanced classification, in which there are too few examples of the minority class for a model to learn the decision boundary effectively [40]. The basic idea of SMOTE is to select samples close to the feature space, and create new samples along that line. After applying SMOTE, the number of near misses and nonconflict cases becomes the same. Tenfold cross validation is also applied when training the LGBM models to avoid overfitting. Two test cases are created to elaborate on the influence of SMOTE: 1) train models without applying SMOTE on the training dataset; and 2) train models after applying SMOTE on the training dataset. The result in Table III shows that all models perform well even without SMOTE. Applying SMOTE to the training data can improve the prediction accuracy for all models.

In our analysis, several input features exhibited strong correlations (correlation coefficient greater than 0.7), such as “speed difference” and “front vehicle speed,” “headway distance” and “front vehicle speed,” and “gaze direction” and “performance.” To manage these highly correlated features, we adjusted our LGBM model by limiting the maximum depth of trees and increasing the number of leaves, thus preventing overly complex models and enhancing flexibility. Alongside these adjustments, we implemented feature preselection to exclude variables with low correlations to the output variable (correlation coefficient smaller than 0.1), specifically focusing on less predictive attributes such as “type 1 (aggressive)” and “type 2 (normal) drivers,” and the “longitude” and “latitude” of the vehicle. This approach ensures that only the most relevant features contribute to model training. Additionally, early stopping was employed to halt training once validation improvements ceased, helping to preserve the model’s generalization capabilities. These strategies collectively enhance our model’s performance, ensuring it captures essential patterns efficiently without being misled by redundant or irrelevant data. Table IV presents the linear correlations between driver behavior features and the predicted risk scores for the next 1 or 2 s. The correlations for driving performance, gaze direction, and EAR are approximately 0.27, indicating weak positive correlations with the risk scores. The MAR metric, however, exhibits a very weak negative correlation, suggesting almost no relationship with the risk scores. Although each feature individually shows a weak correlation, they can still be valuable when used together in a predictive model. LGBM can exploit complex nonlinear interactions and patterns among these features to improve predictive accuracy.

The feature importance of the LGBM models is shown in Fig. 19. The speed difference to the front vehicle, headway distance, Vehicle speed, acceleration, and the angle to the front vehicle play the most important roles when predicting whether there will be a conflict or not. Driver behaviors, such as

**Table V: Performance Comparison Between LGBM Models**

| Case | Evaluation Metric | TTC Next 1 s | TTC Next 2 s | MTTC Next 1 s | MTTC Next 2 s | DRAC Next 1 s | DRAC Next 2 s |
|------|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| i    | Accuracy          | 0.990        | 0.991        | 0.978        | 0.953        | 0.971        | 0.945        |
| ii   | Accuracy          | 0.999        | 0.998        | 0.990        | 0.974        | 0.978        | 0.972        |
| i    | Precision         | 0.992        | 0.992        | 0.977        | 0.954        | 0.970        | 0.948        |
| ii   | Precision         | 0.999        | 0.998        | 0.989        | 0.978        | 0.977        | 0.969        |
| i    | Recall            | 0.992        | 0.993        | 0.977        | 0.955        | 0.970        | 0.944        |
| ii   | Recall            | 0.999        | 0.998        | 0.989        | 0.977        | 0.981        | 0.972        |
| i    | F1-score          | 0.993        | 0.993        | 0.977        | 0.954        | 0.971        | 0.943        |
| ii   | F1-score          | 0.999        | 0.998        | 0.991        | 0.974        | 0.978        | 0.966        |
human factors on the driver can be used to support other studies, such as the impact of system indices are highly correlated to headway distance and speed difference to the front vehicle. However, the roadside camera may lose track of the car when the car is covered by trees or buses. In this study, we developed the ESRM by integrating driver behavior and real-time traffic data to assess roadway risks. Through an Android app, we gathered detailed behavioral data from drivers, supplemented by traffic information from roadside cameras. This combination facilitated a predictive model that successfully calculates risk scores for road segments, providing valuable risk heat maps for traffic authorities to identify safe corridors and enhance trip planning. Despite the model’s strengths, we identified several areas for improvement. Issues with network connections and latency impact the real-time accuracy and efficiency of data collection. Future efforts will focus on expanding the data collection process to include iOS users and refining data mining methods and modeling techniques. Particularly, we plan to replace the current flat model structure in LGBM with a hierarchical model to improve accuracy and model performance. A significant focus will be placed on refining the weight assignment in the fuzzy logic component of our model. The initial weight settings, based on evenly distributed quantiles of crash data, will be revisited. We aim to employ more robust statistical methods such as cluster analysis to empirically validate and adjust these weights according to the severity and economic impact of crashes. This method will ensure that the weights reflect more accurate risk assessments and improve the model’s applicability. Moreover, the categorization of crash thresholds into low, medium, and high levels will be enhanced using statistical techniques to find natural breakpoints in crash data. This will allow for a more precise definition of risk bands, making the model more reliable for safety analysis. Moving forward, these improvements will bolster the model’s precision and relevance, making it a more effective tool for enhancing roadway safety.

### VI. CONCLUSION

Table VI presents the results of expected number of crashes for each road segment as shown in Fig. 3. The results are calculated from the NB model presented in Section III-A. Based on the predicted safety indices for individual drivers and the expected number of crashes based on road segments, risk scores for road segments can be calculated. Confusion matrices in Tables VII and VIII are used to assess the accuracy of predicting risk score levels in the next 1 or 2 s. The results indicate that the model is able to correctly classify risk score levels in the majority of cases. Furthermore, Table IX provides additional evaluation metrics to further evaluate the accuracy of the risk score prediction results.

| Case     | Accuracy | Precision | Recall | F1 Score |
|----------|----------|-----------|--------|----------|
| Next 1 s | 0.922    | 0.831     | 0.854  | 0.779    |
| Next 2 s | 0.855    | 0.773     | 0.766  | 0.693    |
[8] U.S. Department of Transportation Federal Highway Administration, Safety Performance Function Development Guide: Developing Jurisdiction-Specific SPF’s, Univ. North Carolina Highway Saf. Res. Center, Chapel Hill, NC, USA, 2013.

[9] J. Jia et al., “Derivation of the empirical Bayesian method for the Negative binomial-Lindley generalized linear model with application in traffic safety,” Accident Anal. Prev., vol. 170, 2022, Art. no. 106638.

[10] U.S. Department of Transportation Federal Highway Administration, “Crash modification factors clearinghouse,” 2017. Accessed: Jun. 10, 2021. [Online]. Available: https://www.cmfclearinghouse.org/

[11] J. Jia et al., “Safe route mapping of roadways using multiple sourced data,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 4, pp. 3169–3179, Apr. 2022.

[12] C. Han et al., “Modeling traveler’s speed-route joint choice behavior with heterogeneous safety concern,” Anal. Methods Accid. Res., vol. 37, 2023, Art. no. 100253.

[13] R. Srinivasan and K. Bauer, “Safety performance function development guide: Developing jurisdiction-specific SPF’s,” FHWA-SA-14-005, Office of Safety, Federal Highway Administration, USA, 2013.

[14] H. Huang et al., “A multivariate spatial model of crash frequency by transportation modes for urban intersections,” Anal. Methods Accid. Res., vol. 14, pp. 10–21, 2017.

[15] A. Theofillous et al., “Predicting road accidents: A rare-events modeling approach,” Transp. Res. Procedia, vol. 14, pp. 3399–3405, 2016.

[16] L. Zheng and T. Sayed, “Comparison of traffic conflict indicators for crash estimation using peak over threshold approach,” Transp. Res. Rec., vol. 2673, no. 5, pp. 493–502, 2019.

[17] SWOV Institute for Road Safety Research, Naturalistic Driving: Observing Everyday Driving Behavior. The Hague, Netherlands: SWOV, 2012.

[18] F. Guo and Y. Fang, “Individual driver risk assessment using naturalistic driving data,” Accid. Anal. Prev., vol. 61, pp. 3–9, 2012.

[19] National Highway Traffic Safety Administration, “The 100-car naturalistic driving study,” U.S. Dept. Transp., Springfield, VA, USA, 2006.

[20] S. Ren et al., “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[21] W. Liu et al., “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 21–37.

[22] J. Redmon et al., “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 779–788.

[23] T. T. Zhang et al., “A longitudinal scanline based vehicle trajectory reconstruction method for high-angle traffic video,” Transp. Res. C Emerg. Technol., vol. 103, no. C, pp. 104–128, 2019.

[24] R. Jabbar et al., “Real-time driver drowsiness detection for android application using deep neural networks techniques,” in Proc. 9th Int. Conf. Ambient Syst., Netw., Technol., Porto, Portugal, 2018, pp. 400–407.

[25] T. Baltrusaitis, P. Robinson, and L.-P. Morency, “OpenFace: An open source facial behavior analysis toolkit,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Lake Placid, NY, USA, 2016, pp. 1–10.

[26] H. Cai and Y. Lin, “Modeling of operators’ emotion and task performance in a virtual driving environment,” Int. J. Hum. Comput. Stud., vol. 69, no. 9, pp. 571–586, 2011.

[27] P. Ekman and E. L. Rosenberg, What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS), London, U.K.: Oxford Univ. Press, 1997.

[28] D. Borza, R. Itu, and R. Danescu, “In the eye of the deceiver: Analyzing eye movements as a cue to deception,” J. Imag., vol. 4, no. 10, 2018, Art. no. 120.

[29] N. Arbabzadeh and M. Jafari, “A data-driven approach for driving safety risk prediction using driver behavior and roadway information data,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 2, pp. 446–460, Feb. 2018.

[30] H. Huang et al., “Travel route safety estimation based on conflict simulation,” Accid. Anal. Prev., vol. 171, 2022, Art. no. 106666.

[31] G. Guido, F. Saccomanno, A. Vitale, V. Astarita, and D. Festa, “Comparing safety performance measures obtained from video capture data,” J. Transp. Eng., vol. 137, no. 7, 2011.

[32] G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., Long Beach, CA, USA, 2017.

[33] T. T. Zhang et al., “Longitudinal-scanline-based arterial traffic video analytics with coordinate transformation assisted by 3D infrastructure,” Transp. Res. Board Rec., vol. 2675, no. 3, pp. 338–357, 2020.

[34] T. Baltrusaitis et al., “OpenFace 2.0: Facial behavior analysis toolkit,” in Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG), 2018, pp. 59–66.

[35] S. P. Lloyd, “Least squares quantization in PCM,” IEEE Trans. Inf. Theory, vol. 28, no. 2, pp. 129–137, Mar. 1982.