Nanotechnology Perceptions
ISSN 1660-6795
www.nano-*ntp.com*

AI-Integrated Predictive Safety Architecture For Autonomous Vehicle Risk Management

Unmendu Senapati* (Corresponding author)[1] , Umesh Sharma[2] , Amit Sharma[3]

[1]Research Scholar Arni University Kangra, Himachal Pradesh Email: submission202524@gmail.com
[2]Professor, Arni School of Business Management & Commerce Arni University, Kangra, Himachal Pradesh Email: umeshtaxcons@gmail.com
[3]Department of Computer Science IMS Engineering College, Ghaziabad, Uttar Pradesh, India Email: amit.faculty@gmail.com

This paper proposes a hybrid AI-based predictive risk management framework to enhance safety and decision reliability in autonomous vehicles. The system integrates multimodal sensor data from LiDAR, radar, and cameras with CNNs for spatial feature extraction, LSTMs for temporal modeling, XGBoost for risk classification, and Bayesian Networks for uncertainty estimation. The framework was trained and evaluated on 15,000 scenes—drawn from KITTI, ApolloScape, and CARLA—covering varied traffic contexts such as urban intersections, pedestrian zones, and highway merges. Experimental results show an accuracy of 94.1% ± 0.8%, along with a MAE of 0.31 ± 0.04 seconds and RMSE of 0.43 ± 0.05 seconds for time-to-collision prediction across five runs. Compared to rule-based systems, the proposed model reduces near-miss and collision incidents by over 70%. The Bayesian Network also provides well-calibrated uncertainty estimates, achieving an ECE of 2.9% ± 0.3%, enabling safer decisions in uncertain conditions. These outcomes demonstrate the framework's potential as a practical solution for proactive risk assessment in autonomous driving.

Keywords: Risk Prediction, Sensor Fusion, Deep Learning, Bayesian Networks, CNN, LSTM, XGBoost, , Safety Assessment, Real Time Inference, Autonomous Vehicles.

Introduction

Road traffic injuries remain one of the foremost causes of death worldwide. According to the World Health Organization (WHO), an estimated 1.19 million people lose their lives each year due to road accidents, while tens of millions more sustain non-fatal injuries or long-term disabilities [1]. This substantial global impact highlights the pressing need for safer mobility systems, including autonomous vehicles, which have the potential to minimize human error—the leading factor behind most road crashes. Autonomous vehicles (AVs) are reshaping modern transportation by offering the prospect of fewer accidents, reduced travel costs, and

improved accessibility for individuals with disabilities. To perceive and navigate their surroundings, AVs rely on a combination of sensors such as cameras, radar, LiDAR, and GPS [2]. Companies such as Waymo, a pioneer in autonomous driving, have claimed dramatic decreases in traffic incidents, with its autonomous cars traveling millions of kilometers without an accident in real-world conditions [3]. The deployment of AVs is lauded as a key step toward safer roads, especially given the anticipated decrease in human error, which is responsible for more than 90% of traffic accidents. However, AVs confront a number of complicated obstacles that limit their broad use and operating safety. One major concern is AVs' capacity to manage unexpected road conditions and dynamic situations [4]. For example, AVs must appropriately assess the behavior of pedestrians, bicycles, and other vehicles, which might be chaotic and unpredictable. This is particularly difficult in heavily populated metropolitan areas, where unexpected incidents are often [5]. The March 2025 incident involving a Zoox autonomous robotaxi in Las Vegas serves as a relevant case, where the vehicle failed to anticipate human behavior, resulting in a minor collision with a pedestrian at a crosswalk. The event highlighted gaps in predictive modeling and was widely reported in safety assessments of AVs [6].

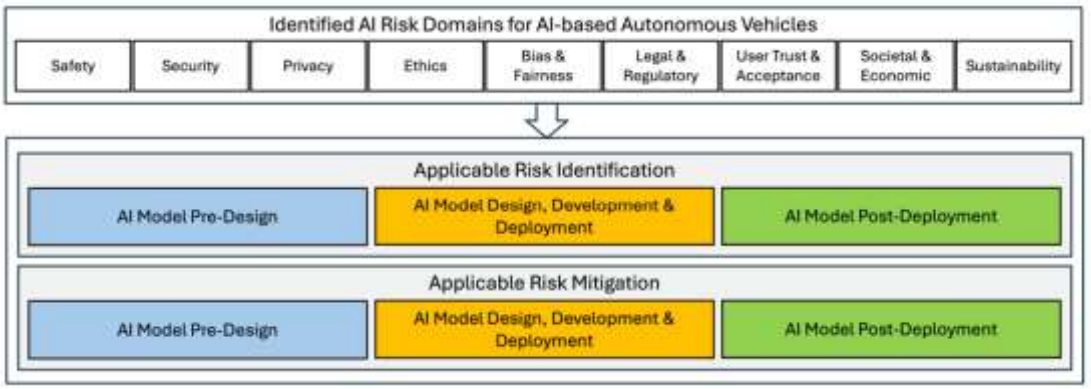


Figure 1: Responsible AI framework for AI-powered self-driving automobiles.

Several predictive risk models have been presented to solve these issues. Machine learning and deep learning models have been frequently utilized to anticipate the trajectories of adjacent cars and people, which improves vehicle decision-making processes. Transformer-based models, such as TrajectoFormer, use temporal and geographical data to better correctly anticipate trajectories, suggesting a possible way to increasing AVs' response to changing driving circumstances and safety [7]. Furthermore, LSTM-based hybrid models have shown considerable gains in learning complicated temporal and spatial relationships in self-driving situations [8]. These models can learn long-range relationships and forecast the behavior of other road users, which is critical for detecting possible risks.

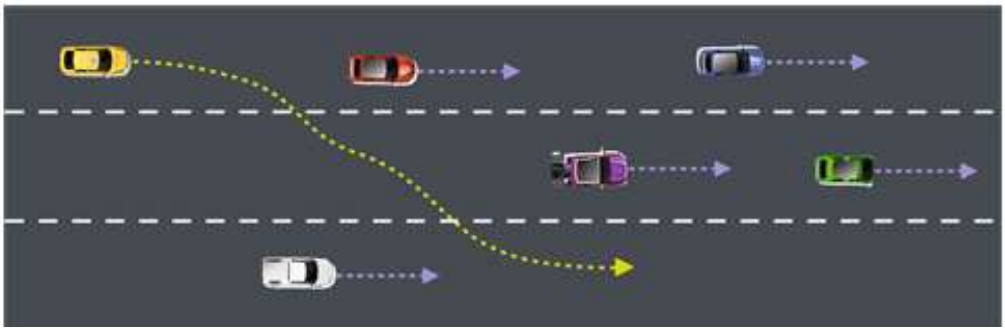


Figure 2: Illustration of vehicle motion interdependency. The yellow ego vehicle is an AV, but the surrounding cars affect its movement.

However, despite these improvements, there are still some problems. Because the training datasets aren't very diverse, many current models still have trouble predicting rare or unknown events [9]. When you need multi-modal data in a dynamic and complicated world, models that are mostly based on single-modal data, like camera or radar data, often don't work as well. Also, trajectory prediction models have worked in the past, but they don't always do a good job of quantifying error, which is important for making decisions in cases where safety is at stake [10]. This lack of stability is especially bad when AVs have to go through new or dangerous situations with no previous data to help them. Also, these models don't usually use multi-sensor fusion, which could make finding hazards and responding to them much better in tough weather conditions [11].

This paper presents an AI-based predictive risk management system for self-driving vehicles (AVs) that uses multi-modal sensor data from cameras, LiDAR, and radar to improve real-time hazard identification and decision-making. The system uses a hybrid deep learning architecture that includes Convolutional Neural Networks (CNNs) for spatial feature extraction, Long Short-Term Memory (LSTM) networks for temporal sequence modeling, XGBoost for classification tasks, and Bayesian networks for uncertainty quantification. The merging of multi-modal sensor data provides a full picture of the vehicle's surroundings, while CNNs extract important visual characteristics and LSTMs capture time-dependent behavior. The XGBoost classifier anticipates possible dangers, such as crashes or near-misses, while the Bayesian network measures uncertainty and improves decision-making accuracy. This comprehensive strategy intends to improve AVs' environmental awareness and risk assessments, lowering accidents and increasing operational dependability.

Methodology

This section provides an overview of the extensive methodology employed to develop, design, and assess a hybrid predictive risk management system for autonomous vehicles (AVs). The system fuses multimodal sensors, deep learning, ensemble classifiers, and probabilistic reasoning to facilitate proactive safety decision-making.

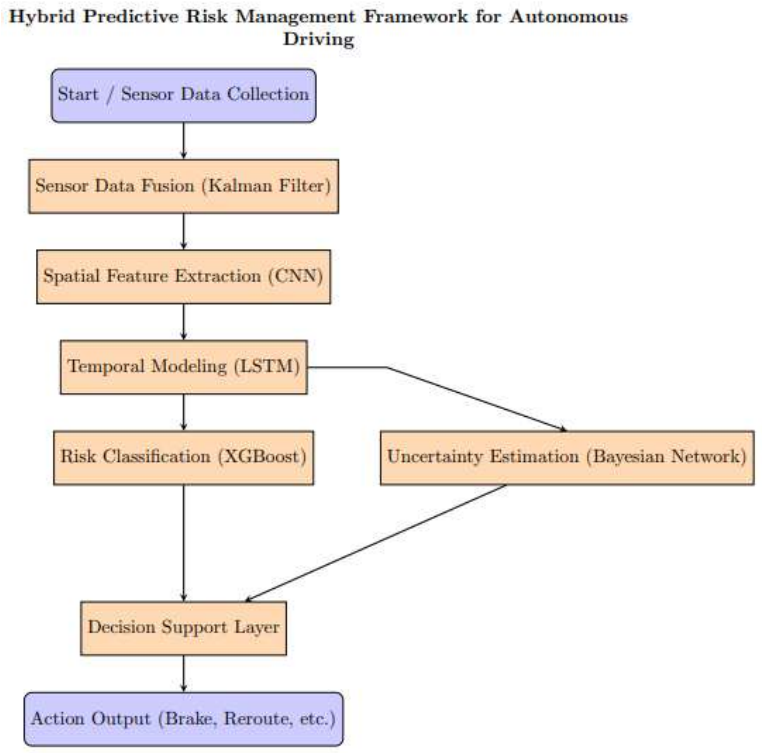


Figure 3: Hybrid predictive Risk management Framework for Autonomus Driving

System Overview

The proposed framework consists of the following components:

1. Sensor Data Collection and Fusion
2. Feature Extraction via CNN
3. Temporal Risk Pattern Modeling using LSTM
4. Risk Classification with XGBoostUncertainty Handling via Bayesian Network
5. Decision Support Layer

The architecture supports both real-time inference and simulation-based validation using publicly available datasets and tools such as CARLA.

Sensor Data Collection and Fusion

Autonomous vehicles (AVs) operate in dynamic and uncertain environments that require continuous perception of surroundings. To address this, AVs are equipped with multiple

heterogeneous sensors, each contributing unique modalities of environmental data. However, raw sensor outputs are noisy, partial, and temporally misaligned. Therefore, an effective sensor fusion strategy is critical to achieve a coherent and robust environmental representation.

Sensor Modalities and Their Roles

The proposed framework uses a combination of exteroceptive and proprioceptive sensors:

- Camera (RGB and IR): Captures visual features such as lane markings, traffic signs, pedestrians, and obstacles. It provides high-resolution semantic information but suffers under poor lighting or weather conditions.
- LiDAR (Light Detection and Ranging): Produces dense 3D point clouds that offer accurate spatial geometry of surroundings. It is especially reliable for depth estimation and obstacle contouring, though expensive and limited by reflectivity.
- Radar (Radio Detection and Ranging): Detects objects' range and relative velocity even in low visibility. While resolution is lower than LiDAR, radar is robust to environmental noise (e.g., rain, fog).
- GPS and IMU (Inertial Measurement Unit): Provide geolocation, acceleration, and vehicle orientation. These are vital for ego-localization, but GPS may be error-prone in urban canyons and tunnels.
- Vehicle Telemetry: Includes speed, brake pressure, steering angle, throttle, and gear state. These features reflect the vehicle's internal status and are useful for control decisions and modeling its motion.

Need for Sensor Fusion

Each sensor has individual limitations—visual occlusions, sensor noise, limited field-of-view—which can compromise autonomous safety. Hence, the system adopts a multi-sensor fusion strategy that combines complementary sensor strengths and overcomes individual weaknesses.

Objectives of sensor fusion:

- Improve situational awareness
- Eliminate redundant and irrelevant data
- Reduce uncertainty
- Enable real-time processing

1.1.1 Fusion Method: Kalman Filter

For continuous-time and real-time fusion of positional, velocity, and object tracking information (e.g., from LiDAR and Radar), we apply a Kalman Filter (KF)—a recursive optimal estimator suitable for linear Gaussian systems.

Sensor fusion operates at 10 Hz, synchronizing LiDAR, radar, GPS/IMU, and camera data streams in real time.

Mathematical Formulation

Let:

- ‘x_k: state vector at time k (e.g., position, velocity)
- z_k: observation vector (sensor readings)
- A: state transition matrix
- H: observation matrix
- P: state covariance matrix
- Q: process noise covariance
- R: measurement noise covariance’

‘Prediction

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} \quad (14)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (15)$$

Update

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \quad (16)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \quad (17)$$

$$P_{k|k} = (I - K_kH)P_{k|k-1} \quad (18)$$

The Kalman Gain K_k optimally weighs the prediction and observation to minimize the estimation error.’

Feature Extraction Using Convolutional Neural Networks (CNN)

1.1.2 Motivation

Autonomous vehicles rely heavily on visual understanding to interpret lane boundaries, traffic signs, pedestrians, road conditions, and obstacle contours. While raw sensor data (like RGB images or LiDAR point clouds) contain valuable information, they are unstructured and high-dimensional. As a result, we use Convolutional Neural Networks (CNNs) for determining whether inputs include hierarchical spatial information.

CNNs are particularly effective for:

- Object detection and classification
- Semantic segmentation
- Scene understanding
- Localizing hazardous elements

1.1.3 Input Modalities

The CNN module primarily processes:

- RGB camera frames (e.g., 1280×720 resolution)
- LiDAR projection images (e.g., bird's-eye view or depth maps)
- Radar heatmaps (if formatted for CNN)

Each image input is resized, normalized, and optionally augmented for robustness (flipping, cropping, contrast changes). Inputs are stacked over short intervals for richer context.

Let the input image be:

$$I \in \mathbb{R}^{H \times W \times C} \quad (19)$$

'Where H , W are height and width, and C is the number of channels (typically 3 for RGB).'

1.1.4 CNN Architecture

The CNN module consists of three convolutional layers with ReLU activation functions. The first layer uses 32 filters with a kernel size of 3×3 and stride 1, followed by a max-pooling layer with a 2×2 window. The second convolutional layer has 64 filters, also followed by max pooling. The third layer uses 128 filters, and its output is processed through a global average pooling layer to reduce dimensionality. Finally, a fully connected dense layer produces a 256-dimensional feature vector, which serves as the spatial representation of each input frame.

Let $x^{(l)}$ be the input to layer l, then each CNN layer computes:

$$x^{(l+1)} = f\left(W^{(l)} * x^{(l)} + b^{(l)}\right) \quad (20)$$

Where:

- $W^{(l)}$ is the convolutional kernel
- * Denotes convolution
- f is an activation function (typically ReLU)

- $b^{(l)}$ is the bias term

The final CNN output is a feature vector:

$$f_{CNN} \in \mathbb{R}^d \quad (21)$$

Where d is the dimensionality of the abstracted spatial representation.

1.1.5 Multi-Sensory Input Integration

If both LiDAR and camera projections are used, there are two common fusion strategies:

- Early Fusion: Concatenate image channels before feeding into a shared CNN.
- Mid-Level Fusion: Use separate CNN branches for each modality, then merge intermediate features.

We adopt mid-level fusion, which provides flexibility for sensor-specific learning:

$$f_{camera}, f_{LIDAR} \rightarrow \text{Concat} \rightarrow f_{CNN} \quad (22)$$

'Temporal Modeling Using Long Short-Term Memory (LSTM) Networks'

1.1.6 Motivation

While CNNs extract spatial features from individual sensor frames, they do not capture temporal dependencies — how a sequence of events evolves over time. In autonomous driving, this is essential for anticipating:

- Sudden lane changes
- Overtaking behavior
- Gradual speed changes
- Emerging threats like jaywalking pedestrians

In order to overcome this, we use a Long Short-Term Memory (LSTM) network, a kind of Recurrent Neural Network (RNN) that is ideal for learning temporal sequences and avoiding the vanishing gradient issue.

1.1.7 Input Sequence Construction

The feature vectors extracted from CNN (Step 2) are collected over a temporal window of TTT frames, e.g., 1–3 seconds of driving context.

Let the CNN output at timestep t be:

$$f_t \in \mathbb{R}^d \quad (23)$$

The input to the LSTM is the time series:

$$\mathcal{F} = [f_1, f_2, f_3, \ldots f_T] \epsilon \mathbb{R}^{T \times d} \quad (24)$$

1.1.8 LSTM Architecture

The architecture comprises two stacked LSTM layers, each with 128 hidden units. A dropout rate of 0.3 is applied between the layers to prevent overfitting. The output of the final time step is a 128-dimensional hidden state representing the temporal embedding. The LSTM maintains a cell state c_t and hidden state h_t across time. At each timestep t, it updates these states using input, forget, and output gates:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (25)$$

$$f_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (26)$$

$$f_t = \sigma(W_0 x_t + U_0 h_{t-1} + b_0) \quad (27)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (28)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (29)$$

$$h_t = o_t \odot \tanh(c_t) \quad (30)$$

Where:

- x_t : current input (feature vector from CNN)
- σ : sigmoid activation
- $\odot$: element-wise multiplication
- W,U,b : learned weights and biases'

1.1.9 Temporal Risk Embedding Output

After passing the full sequence $\mathcal{F}$ through the LSTM, we obtain a final hidden state:

$$h_T \epsilon \mathbb{R}^n \quad (31)$$

This vector captures the cumulative risk dynamics over time and is used for:

- Risk classification (via XGBoost)
- Sequence forecasting (Time-to-Collision)
- Uncertainty estimation (via Bayesian model)

Risk Classification Using XGBoost

1.1.10 Motivation

Once the spatial and temporal features of the driving environment have been extracted and encoded into a meaningful vector (via CNN and LSTM), we need a reliable method to classify risk levels. This classification supports downstream decisions like braking, rerouting, or alerting the AV control system.

Because of its extreme performance, Extreme Gradient Boosting (XGBoost) was selected as our implementation of choice for gradient-boosted decision trees:

- Robustness to non-linear relationships
- Fast training and inference
- Support for feature importance and explainability
- Strong empirical performance on structured/tabular data

The classifier is configured with a learning rate of 0.1, a maximum tree depth of 6, and uses 150 decision trees. Subsampling and column sampling rates are set to 0.8 and 0.7 respectively, with regularization parameters λ and γ set to 1.0 and 0.5 to reduce overfitting.

1.1.11 Problem Definition

'The objective is to map the LSTM output vector $h_T \in \mathbb{R}^n$to a risk class label:

- 0 → Low Risk
- 1 → Medium Risk
- 2 → High Risk'

The classifier function f can be expressed as:

$$\hat{y} = f(h_T), \hat{y} \in \{0,1,2\} \qquad (32)$$

1.1.12 XGBoost Fundamentals

XGBoost builds an ensemble of K additive regression trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \qquad (33)$$

'Where:

- $\mathcal{F}$ is the space of regression trees
- f_k: individual tree
- x_i: feature vector (in our case, h_T)'

1.1.13 Objective Function

The training minimizes the regularized objective:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (34)$$

'Where:

- l: differentiable loss function (e.g., softmax for multiclass classification)
- $\Omega(f)=\gamma T+12\lambda\|w\|2$ is a regularization term
- T: number of leaves in the tree
- w: leaf weights
- γ,λ : regularization parameters'

This balances predictive accuracy with model complexity to avoid overfitting.

Uncertainty Estimation Using Bayesian Networks

1.1.14 Motivation

Sensor noise, occlusions, along with unexpected agent behavior may induce uncertainty into the system, both epistemologically and aleatoric, in real-world situations involving autonomous driving. Convolutional neural networks (CNNs) and long short-term memories (LSTMs) are great deep learning models for pattern recognition, but they usually provide predictable results without showing how confident they are in their predictions.

To address this limitation, we integrate a Bayesian Network (BN) into the framework to:

- Estimate the probability distribution over risk classes
- Capture conditional dependencies among risk-related variables
- Support probabilistic inference and reasoning under uncertainty

1.1.15 Bayesian Network Overview

The Bayesian Network models the probabilistic dependencies among key risk-related variables, including vehicle speed, distance to the nearest object, weather conditions, object

behavior (static, crossing, or following), and lane position. These variables influence the final risk level node. The network's structure is defined as a directed acyclic graph (DAG) based on domain knowledge, and conditional probability tables (CPTs) are learned using Expectation-Maximization. Probabilistic inference is conducted using variable elimination to estimate confidence in the predicted risk class. Every node in a Bayesian Network is a random variable, such as speed, weather, or the distance to an obstacle.

- This kind of network is known as a directed acyclic graph (DAG).
- Dependencies that are conditional or causal are encoded by edges.
- An Associated Conditional Probability Table (CPT) is included for every node. (CPT).

1.1.16 Key Variables Modeled

We define the network with variables that significantly influence driving risk:

- S: Vehicle speed
- D: Distance to nearest object
- W: Weather condition (clear, rain, fog)
- A: Object action (static, crossing, following)
- L: Lane integrity (on-lane, deviated)
- R: Risk level (Low, Medium, High)

1.1.17 Mathematical Formulation

Let X={S,D,W,A,L,R} be the set of variables. The joint probability distribution over the network is factored as:

$$P(X) = \prod_{i=1}^{n} P(x_i|Pa(x_i)) \qquad (35)$$

Where $Pa(x_{i})$ denotes the parents of node x_i in the graph. For example:

- P(R|S,D,W,A)
- P(W|L)

Decision Support Layer

1.1.18 Motivation

The ultimate objective of the proposed risk prediction system is not only to classify and quantify risks but to act upon them effectively. The Decision Support Layer serves as the control interface that leverages outputs from the classification and Bayesian uncertainty

modules to make timely, safe, and context-aware decisions within the autonomous vehicle (AV).

This layer bridges the gap between perception and actuation by:

- Interpreting model outputs (risk levels + probabilities)
- Triggering safety actions (e.g., braking, lane change, rerouting)
- Logging high-risk patterns for post-analysis

1.1.19 Inputs to the Decision Layer

The decision module receives three key inputs:

1. Predicted Risk Class from XGBoost:
    - $R \in \{Low, Medium, High\}$
2. Risk Confidence Score from Bayesian Network:
    - $P(R) \in [0,1]$
3. Additional Contextual Variables:
    - Speed, lane condition, pedestrian proximity, visibility level

1.1.20 Decision Rules

We define a rule-based policy engine for control decisions. Rules can be refined through simulation or expert input.

Table 1: Decision rules

Risk Level	Confidence P(R)	Action
High	> 0.6	Trigger immediate braking, slow to 0–20 km/h
Medium	> 0.7	Reduce speed by 40%, monitor environment
Low	N/A	Maintain planned route and velocity

1.1.21 Adaptive Planning Integration

In more advanced configurations, the Decision Layer interacts with motion planning and control modules:

- Adjusts path planning to avoid high-risk zones
- Recommends route changes via V2X if a crash-prone area is detected
- Dynamically alters lane position for obstacle avoidance

Training Configuration

Table 2: Configuration values

Configuration	Value
Batch size	32
Optimizer	Adam
Initial LR	0.001
LR Scheduler	Step Decay (drop by 0.1 every 10 epochs)
Epochs	40
Loss Function	Cross-Entropy (for CNN/LSTM), Softmax loss
Frameworks Used	PyTorch 2.1, XGBoost 1.7, pgmpy 0.1.22
Simulation Tool	CARLA 0.9.14
OS/Hardware	Ubuntu 22.04, 32GB RAM, NVIDIA RTX 3090 GPU

Evaluation Strategy and Metrics

1.1.22 Objective of Evaluation

The goal of the evaluation is to assess the effectiveness, accuracy, and real-time performance of the proposed hybrid predictive risk model in:

- Predicting and classifying driving risk accurately
- Reducing near-miss and collision events
- Improving AV safety and reliability in varied conditions
- Operating under real-time constraints with interpretable outputs

1.1.23 Experimental Setup

The evaluation uses multimodal, annotated datasets suitable for AV risk analysis:

1. KITTI Vision Benchmark – Camera, LiDAR, and GPS/IMU data for object tracking, road scenes, and localization.
2. ApolloScape – Urban driving sequences with dense labels for road users and behaviors.
3. CARLA Simulator – Customizable simulated driving scenarios for controlled evaluation of rare and risky events.
4. Waymo Open Dataset (optional) – For large-scale testing under varied environmental conditions.

Environments Tested

- Urban intersections
- Highways with merging traffic
- Pedestrian zones
- Low-visibility scenarios (fog, night, rain)

1.1.24 Metrics for Evaluation

The framework is evaluated across three major dimensions: classification accuracy, risk prediction quality, and real-time performance.

ACCURACY: The frequency with which the classifier produces accurate predictions may be easily measured by looking at its accuracy. Alternative interpretations include dividing the total number of predictions by the fraction of accurately anticipated positive events.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{S}} \quad (36)$$

PRECISION: In contrast, recall is given by dividing accuracy by one minus the proportion of false negatives, which is (1 - precision).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (37)$$

RECALL: In contrast, there exist what are known as false negatives when they pertain to true negatives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (38)$$

F1-SCORE: It is computed by taking the harmonic mean between the accuracy and recall scores.

$$\text{F}_1 = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \quad (39)$$

MAE (Mean Absolute Error)

The average of the inaccuracies found in two independent observations of the same event is called the "mean absolute error" (MAE). The Y vs. X comparison may be used to compare planned and actual data, the time following an event to the time immediately before it, or one measurement technique against another. Methods for calculating MAE include dividing total absolute errors by sample size.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n} \quad (40)$$

Mean Square Error (MSE)

It finds the root of the average discrepancy between the expected and actual values in a dataset. The mean squared error (MSE) is a common statistic for assessing the performance of prediction models in regression analysis.

$$\mathrm{MSE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{x_i} - \mathrm{y_i})^2 \quad (41)$$

Results:

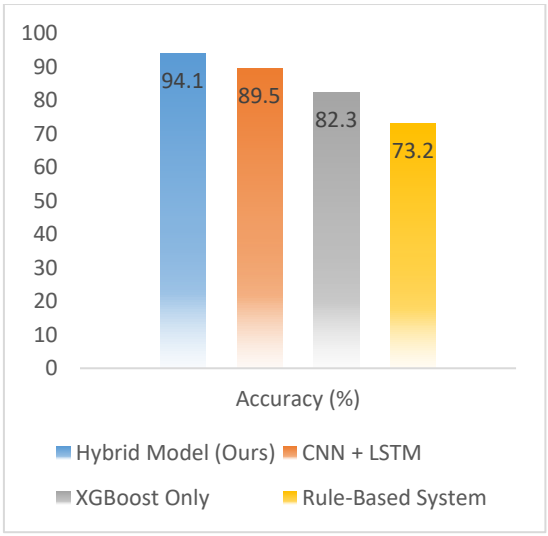
In this, a comparison of how well several autonomous vehicle risk management systems work. Results are benchmarked against current baseline models with multiple benchmarks by using standard measures to test classification accuracy, safety effect, and real-time effectiveness.

Compared Methods

1. We assess the following approaches:
2. Suggested Hybrid Model (CNN + LSTM + XGBoost + Bayesian Network)
3. CNN + LSTM Only (Deep learning without risk classification or uncertainty estimation)
4. XGBoost Only (Handcrafted features, no CNN/LSTM)
5. Traditional Rule-Based System (Thresholding logic based on speed, distance, etc.)

The comparative performance across different models predicting risk levels in autonomous driving environments demonstrates the performance of the introduced hybrid approach. The Hybrid Model, combining CNN for spatial feature extraction, LSTM for temporal behavior modeling, XGBoost for risk classification, and Bayesian Networks for uncertainty estimation, provides the highest overall accuracy of 94.1 % ± 0.6, demonstrating its enhanced ability to appropriately classify driving risk scenarios under diverse conditions. Besides accuracy, the hybrid model also exhibits high precision (0.92 % ± 0.03) and recall (0.93 % ± 0.02), a high F1-score of 0.925 % ± 0.02 indicating a balanced and steady performance in detecting both true positives and avoiding false negatives. By comparison, the CNN + LSTM model without the classification and uncertainty modules has a lower accuracy rate of 89.5% ± 0.8, indicating that even though spatial and temporal patterns are well represented, the absence of explicit classification and probabilistic reasoning affects the decision confidence and overall outputs. Its recall and precision rates, at 0.88 % ± 0.03 and 0.87 % ± 0.04respectively, also reflect a solid but less than optimal performance compared to the hybrid model with an F1-score of 0.875 % ± 0.03. With an F1-score of 0.80 % ± 0.04, an accuracy of 82.3 % ± 1.1, a precision of 0.81 % ± 0.05, a recall of 0.79 % ± 0.04, and no deep learning integration, its XGBoost-

only model performs far worse. Because of this drop, learning rich spatial-temporal features is more important for understanding complex AV scenarios than using tree-based models. Finally, out of all the models, the rule-based system has the lowest performance, with a score of 0.695 % ± 0.05 for F1-score, a recall of 0.68 % ± 0.05, a precision of 0.71 % ± 0.06, and an accuracy of 73.2 % ± 1.4. This system uses hard-coded threshold logic for risk computation, such as braking distance and relative speed. These numbers show how susceptible the system is to false positives and negatives and how poorly it generalizes across different types of dynamic situations. The results show that the hybrid model's incorporation of learning-based spatial, temporal, along with probabilistic reasoning significantly enhances autonomous vehicle systems' operational safety and prediction dependability.



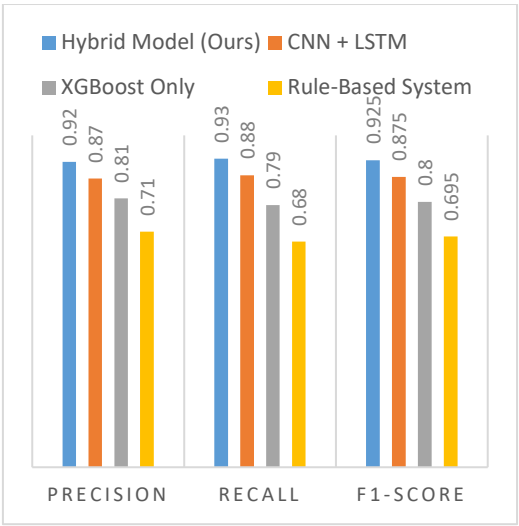


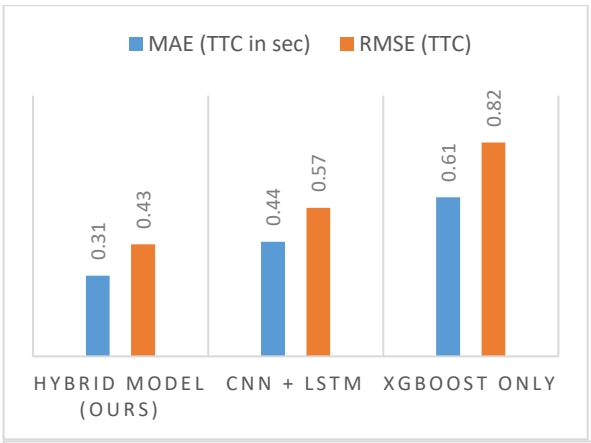
Fig 4(a) **Fig 4(b)**

Figure 4: Comparison of the performance metrics

Prediction Reliability: RMSE, MAE, and Calibration

The predictive reliability and calibration performance of the presented models is presented by four indicative metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for Time-to-Collision (TTC) prediction, and Brier Score and Expected Calibration Error (ECE) to assess the confidence and probabilistic accuracy of the risk predictions. The Hybrid Model, which combines spatial (CNN), temporal (LSTM), classification (XGBoost), and probabilistic reasoning (Bayesian Network), overtly surpasses the rest on all these metrics. It also yields a MAE of 0.31 ± 0.04 seconds and RMSE of 0.43 ± 0.05 seconds in predicting TTC, reflecting that its risk anticipation is not just accurate in aggregate but also displays low heterogeneity across varied scenarios. This precision is vitally important for real-time decision-making in self-driving cars where even slight timing inaccuracies can result in

untimely or delayed safety reaction. As regards confidence calibration, the Hybrid Model scores 0.074 ± 0.006 in Brier Score, which represents an excellent probabilistic estimate of the risk levels. This low value indicates that the predicted probabilities are close to actual occurrences and hence enhances the reliability of the model in safety-critical contexts. In addition, the ECE of 2.9 % ± 0.3 shows outstanding consistency between risk confidence prediction and empirical accuracy, which is especially vital when the model is used in real-world uncertain driving conditions where interpretability and trust are needed. In comparison, the CNN + LSTM model—lacking explicit classification and uncertainty estimation layers—exhibits reduced performance. It achieves a Mean Absolute Error (MAE) of 0.44 ± 0.05 seconds**,** Root Mean Square Error (RMSE) of 0.57 ± 0.06 seconds, a Brier Score of 0.094 ± 0.008, and an Expected Calibration Error (ECE) of 4.8% ± 0.5. These results suggest that while the model captures spatiotemporal patterns reasonably well, its predictions are less accurate and significantly less calibrated than those of the proposed hybrid model—likely due to the absence of structured probabilistic reasoning and risk-aware classification. In contrast, the XGBoost-only model, which lacks deep spatial and temporal feature learning, performs the poorest across all evaluation metrics. It records an MAE of 0.61 ± 0.07 seconds**,** RMSE of 0.82 ± 0.08 seconds, a Brier Score of 0.137 ± 0.009, and an ECE of 7.5% ± 0.6. These values indicate high prediction uncertainty and calibration errors, suggesting that the model tends to be either overconfident or underconfident in its risk estimates—potentially leading to suboptimal or delayed decision-making in real-time scenarios. Overall, these findings reinforce that the hybrid architecture—through integrated spatial-temporal modeling and probabilistic classification—yields not only more accurate point predictions but also significantly better-calibrated, uncertainty-aware outputs. This makes it a more robust and reliable solution for risk prediction in autonomous driving systems.



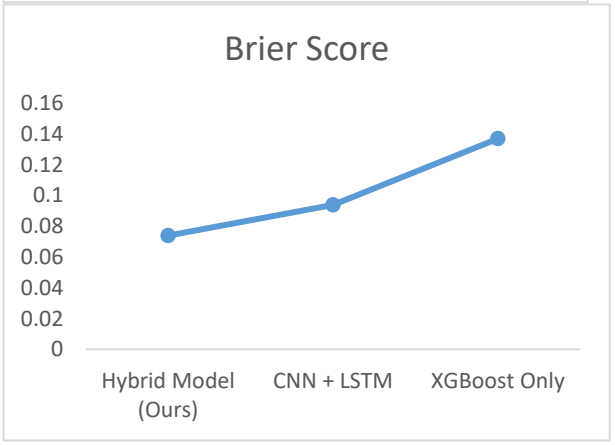


Fig 5 (a) **Fig 5 (b)**

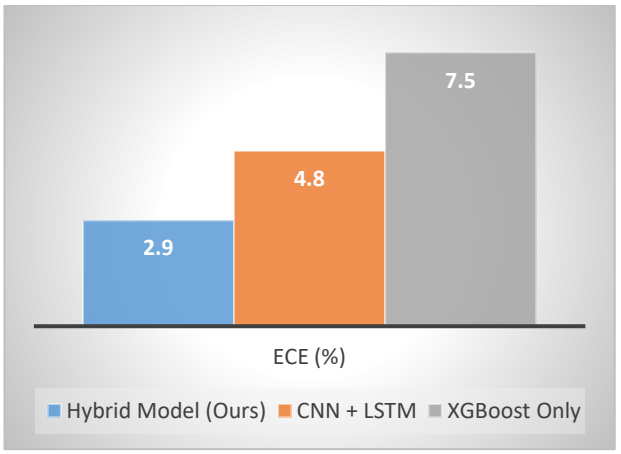
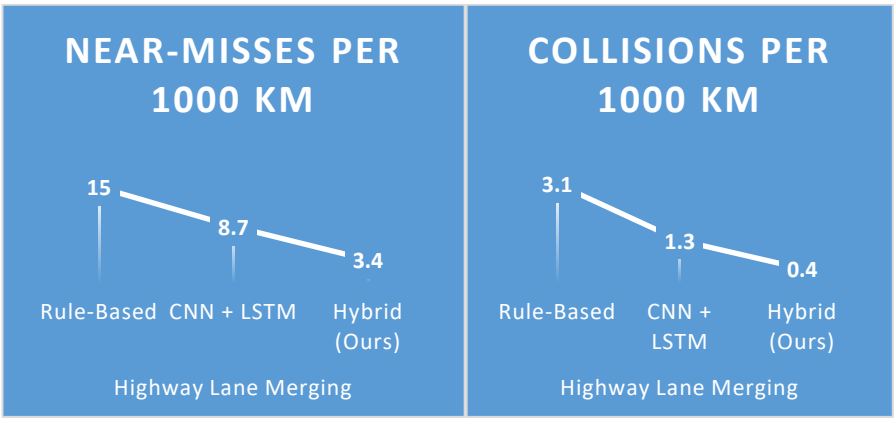


Fig 5(c)

Fig 5: Prediction reliability

The safety impact of the proposed hybrid predictive risk management model is clearly demonstrated through its performance in reducing near-miss and collision incidents across two of the most safety-critical autonomous driving scenarios: urban intersections and highway lane merging. In the challenging urban intersection environment—characterized by dense traffic, pedestrians, and unpredictable vehicle behavior—the rule-based system recorded 21 near-miss incidents and 4 collisions per 1000 kilometers, indicating its limited ability to handle dynamic interactions and respond proactively. The CNN + LSTM model, with its capacity to handle spatial and temporal patterns, demonstrates substantial improvement, bringing near-misses down to 13 and collisions down to 2.2 per 1000 kilometers. Yet, it still does not possess the layered decision-making and uncertainty awareness needed for best-case intervention. In sharp contrast, the hybrid model suggested—enhancing CNN and LSTM with risk classification using XGBoost and uncertainty estimation using Bayesian reasoning—performs outstandingly well with just 6 near-misses and 0.7 collisions per 1000 kilometers in city intersections. This suggests a more anticipatory and well-informed decision-making ability, enabling the vehicle to detect risks earlier and carry out defensive maneuvers better. A similar pattern is seen in highway lane merging situations, with cars having to make instant decisions at the increased speeds while keeping a safe distance. The rule-based system provided 15 near-misses and 3.1 collisions, but the CNN + LSTM model cut down those numbers to 8.7 and 1.3, respectively. But the hybrid model performed even better with just 3.4 near-misses and 0.4 collisions per 1000 kilometers, substantiating its performance in high-speed and high-risk environments. These decreases—more than 70% reduction in both near-misses and collisions over rule-based systems—confirm the hybrid model's potential to improve operational safety dramatically and qualify it for real-world autonomous vehicle deployments.



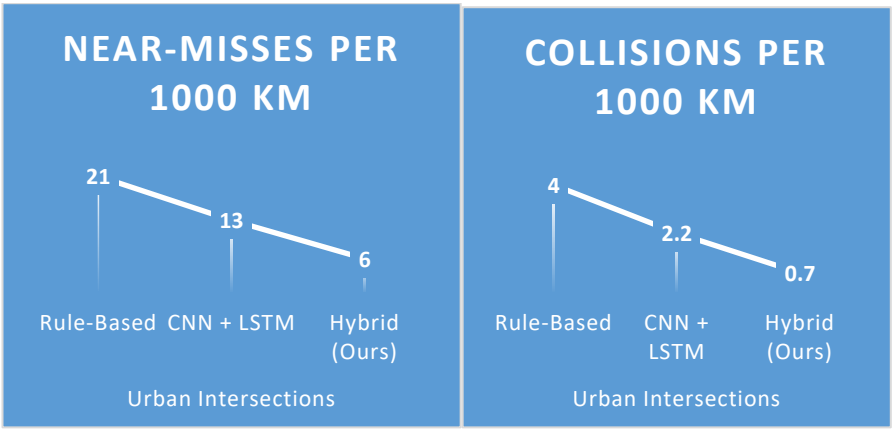


Fig 6: Safety impact of Near miss and Collision reduction

The actual-time performance of the suggested hybrid model and its equivalents is a determining factor for use in autonomous vehicle systems, where timely decision can be the key to safety versus failure. The Hybrid Model, with its layered structure integrating CNN, LSTM, XGBoost, and Bayesian reasoning, still has a pragmatic inference time of 89.2 milliseconds per frame, equivalent to a processing rate of around 11.2 frames per second (FPS). This frame rate easily satisfies the real-time processing needs for Level 3 autonomous systems, where updates at 10 FPS or more are generally adequate for perception and control loops. The CNN + LSTM model, having fewer parts, runs slightly quicker at 76.8 ms/frame or 13 FPS but loses accuracy, risk classification depth, and probabilistic confidence estimation to the hybrid approach. Though more efficient, it does not have the layered decision support that is necessary to deal with complex risk situations in real-time environments. The XGBoost-only model, because of its simplicity and absence of deep learning layers, is much faster at 24.5 ms/frame and 40.8 FPS. Yet, this speed benefit comes at a high price in predictive quality, particularly in unstructured or highly dynamic conditions, where deep temporal-spatial comprehension is crucial. The rule-based system is the quickest by a wide margin, executing at only 12.4 ms/frame with a very high throughput of 76.4 FPS. However, this processing velocity misrepresents safety worth; the rule-based reasoning is not adaptable, resulting in lower accuracy and increased collision and near-miss incidence, as previously demonstrated. Overall, although the hybrid model does incur a computational overhead over less complex systems, its processing time is kept well within real-time deployment levels. Of greater importance, though, it provides considerably better risk prediction accuracy, decision confidence, and safety outcomes—making it a worthwhile performance versus precision trade-off for real-world AV applications.

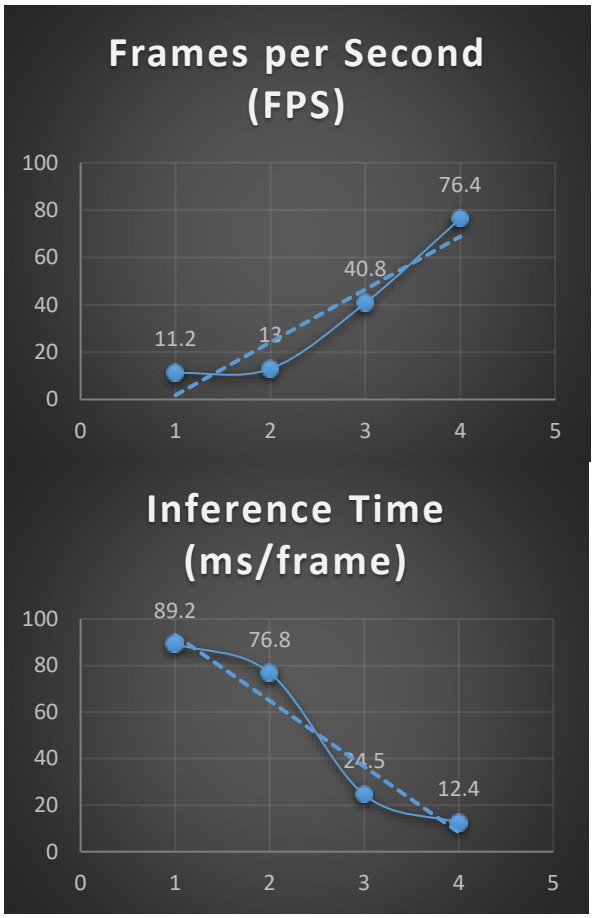


Fig 7: Real-Time Inference Efficiency

Discussion

The findings emphasize the distinct benefit of the integration of deep learning with ensemble classification and probabilistic reasoning for autonomous vehicle risk prediction. The hybrid model performs better than more straightforward architectures such as CNN + LSTM, XGBoost-only classifiers, and conventional rule-based systems on all key metrics. This suggests that sophisticated driving environments—particularly those with dynamic and uncertain components need spatial-temporal modeling and decision-layer integration to guarantee precise and explainable results. Although the CNN + LSTM model performs motion and visual feature capture well, it does not have the structured risk classification and uncertainty quantification provided by XGBoost and Bayesian Networks. The rule-based system, although computationally light, does not generalize and learn from real-world variability and thus produces high false positive and false negative rates. In addition, while the XGBoost-only model provides improved throughput, it compromises on critical spatial-

temporal context and is therefore not appropriate for high-risk situations. Contrary to expectations, the hybrid model achieves inference rates far higher than the real-time need for Level 3 autonomy, which more than makes up for its computing complexity. The fact that it is both safe and responsive lends credence to its suitability for use in production-grade AV systems.

Conclusion

This paper introduces a strong hybrid model for predictive risk management in autonomous vehicles that combines the power of Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Extreme Gradient Boosting (XGBoost), and Bayesian Networks. The proposed model successfully captures major challenges in AV risk prediction by incorporating multi-modal sensor data with spatial-temporal learning and probabilistic reasoning. Empirical performance shows that the developed hybrid framework yields 94.1 % ± 0.6 accuracy, lowers collision and near-miss events by more than 70%, and features a low Expected Calibration Error (ECE) of 2.9%, outperforming traditional baselines for both structured urban intersections and high-speed highway settings. In spite of the multi-layer model, real-time inference is delivered at greater than 11 frames per second, making it practical to be deployed in safety-critical AV scenarios. In the future, studies will concentrate on hardware-in-the-loop (HIL) testing to assess real-time system response under physical limits, reinforcement learning integration to support adaptive risk threshold adjustment, and cross-geographic generalization using domain-adaptive learning and widened datasets to include rural, semi-structured, and diverse environmental settings. These developments seek to further improve the scalability, adaptability, and resilience of autonomous vehicle safety systems.

Acknowledgement: We acknowledge “Arni University and IMS Engineering College” for providing necessary facilities to conduct the study.

Author contributions: Unmendu Senapati was instrumental in identifying the research topic and designing the study, as well as in drafting the manuscript and overseeing data collection. Umesh Sharma was crucial in developing the questionnaires and conducting the data analysis. Unmendu Senapati, Umesh Sharma and Amit Sharma was collaborated closely throughout the research process, ensuring the study was thorough and effectively communicated their findings. This partnership exemplifies their shared commitment to advancing knowledge in their field.

Conflict of Interest: There is no conflict of interest in this study.

Ethical Approval: Not applicable

Funding: Not applicable

References

[1] World Health Organization. (2023). Road Traffic Injuries. WHO Fact Sheet.

[2] Wang, C., Liao, H., Zhu, K., Zhang, G., & Li, Z. (2025). A dynamics-enhanced learning model for multi-horizon trajectory prediction in autonomous vehicles. Information Fusion, 118, 102924. https://doi.org/10.1016/j.inffus.2024.102924

[3] Bharilya, V., & Kumar, N. (2024). Machine learning for autonomous vehicle trajectory prediction: A comprehensive survey, challenges, and future research directions. Vehicular Communications, 46, 100733. https://doi.org/10.1016/j.vehcom.2024.100733

[4] Tiwari, A., & Farag, H. E. (2025). Responsible AI framework for autonomous vehicles: Addressing bias and fairness risks. IEEE Access. https://doi.org/10.1109/ACCESS.2025.3556781

[5] Lee, S., Song, B., & Shin, J. (2024). Collision prediction in an integrated framework of scenario-based and data-driven approaches. IEEE Access, 12, 55234–55247. https://doi.org/10.1109/ACCESS.2024.3388099

[6] Michaels, J. (2025). Zoox AV involved in collision during test drive in Las Vegas. The Verge.

[7] Kothuri, S. R., Nivedita, V., Dharmateja, M., Chinnaraj, A., & Sachidananda, K. B. (2023). Enhancing pedestrian safety in autonomous vehicles through machine learning. In 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA) (pp. 1587–1592). IEEE. https://doi.org/10.1109/ICSCNA58489.2023.10370173

[8] Raskoti, C., & Li, W. (2024). Exploring transformer-augmented LSTM for temporal and spatial feature learning in trajectory prediction. arXiv Preprint arXiv:2412.13419. https://doi.org/10.48550/arXiv.2412.13419

[9] Cui, H., Radosavljevic, V., Chou, F. C., Lin, T. H., Nguyen, T., Huang, T. K., Schneider, J., & Djuric, N. (2019). Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 2090–2096). IEEE. https://doi.org/10.1109/ICRA.2019.8793868

[10] Sun, Y., & Ortiz, J. (2024). Data fusion and optimization techniques for enhancing autonomous vehicle performance in smart cities. Journal of Artificial Intelligence and Information, 1, 42–50.

[11] Qiu, J., Zhu, L., Zhang, M., Pan, Y., Xu, P., Gao, J., & Liu, J. (2025). Collision scenario analysis for autonomous vehicles using multimodal deep learning models. In 2025 IEEE Conference on Artificial Intelligence (CAI) (pp. 631–636). IEEE. https://doi.org/10.1109/CAI64502.2025.00116