RoadSense – An AI-Based Vehicle Alert System Driven by V2V Mesh Communications

Amit Resh, Yigal Hoffner, Dean Itzhak, Tomer Ifargan, Shir Amar
Dept. of Computer Engineering
Shenkar - Engineering. Design. Art.
Ramat-Gan, Israel
e-mail: amit.resh@shenkar.ac.il, yigal.hoffner@shenkar.ac.il, deanitzhak@gmail.com, tomer.ifargan@stu.shenkar.ac.il, shiramar0401@gmail.com

***Abstract*— Vehicle-to-Vehicle (V2V) communication has long been recognized as a key player in cooperative safety, extending a vehicle's hazard awareness beyond the limits of local sensors. However, most prior studies address only one aspect of this challenge, either optimizing communication performance (latency, reliability, congestion) or training decision-making agents. This paper presents *RoadSense*, an integrated V2V hazard detection and response system designed to improve driving safety by extending awareness beyond local sensors using a mesh communication network and an AI-based agent. The AI agent is trained with reinforcement learning on a hybrid dataset of real V2V logs and domain-randomized simulations to predict and respond to potential forward collisions by issuing graded braking warnings.**

Keywords - Vehicle-to-Vehicle (V2V) Communication; Cooperative Collision Avoidance; Multi-agent reinforcement learning (MARL); Hazard Detection; Multi-Hop Mesh Network.

I. INTRODUCTION

Traffic congestion ahead of a vehicle can develop suddenly, leading to rapid deceleration or abrupt stops. Chain-reaction collisions frequently occur when a leading vehicle brakes unexpectedly, and following drivers have insufficient time or context to respond. While the first trailing car may detect and react, those farther behind often respond too late, typically only after the vehicle immediately ahead slows or collides.

A driver's or local controller's perception is inherently limited to what lies within direct view. This limitation worsens in adverse conditions, such as fog, rain, curves, or visual obstructions from trucks or buildings. Modern sensors, cameras, radar, ultrasonic units, and Light Detection & Ranging (LiDAR) extend local awareness but remain bound by physical range and line-of-sight constraints. Consequently, hazards hidden by terrain, traffic, or weather can remain undetected [1].

Vehicle-to-Vehicle (V2V) communication offers a complementary means of overcoming these limitations. By sharing safety-critical motion and hazard data among vehicles, V2V systems extend situational awareness beyond visual perception. Early hazard notifications increase available reaction time for both drivers and automated controllers, thereby reducing collision risk.

As shown in Figure 1, the control loop is extended by integrating information from an ad hoc vehicle-based mesh network. This addition transforms the process from reactive to predictive, enabling the ego-car to anticipate conditions beyond the range of its local sensors.

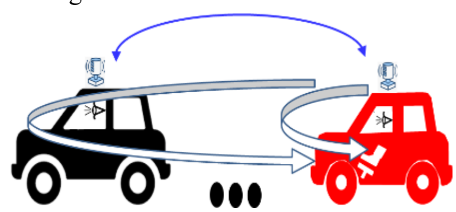

Figure 1. Ego-car extended driver-control loop with V2V mesh communication: data from preceding traffic augments local perception, extending the driver's awareness.

A. The RoadSense Project Goal

This project proposes an integrated hardware-software system that extends a vehicle-based mesh network and an AI-driven decision agent to process real-time data from the host vehicle and surrounding vehicles, including those beyond line of sight, providing timely alerts to the driver to apply brakes and prevent potential forward collisions (Figure 1). By continuously analyzing these inputs, the decision-making process can evaluate evolving traffic conditions and issue real-time alerts regarding potential hazards ahead. This approach enhances driving safety by enabling early detection of potential hazards over extended distances, thereby allowing braking intervention before hazards become critical.

The expected safety improvements are:

- Early alerts: Vehicles share warnings about hazards before drivers or local sensors detect them.
- Faster response: Extra time to react promptly lowers crash risk.
- Extended visibility: Hazards can be detected beyond curves, trucks, hills or other obstructions.

B. The RoadSense Research and Development Approach

Although the *RoadSense* system encompasses all aspects of an in-vehicle hazard detection and response system, the project's focus lies in four interrelated domains:

- Dynamic V2V communication: Developing a flexible ad-hoc wireless mesh that connects moving vehicles without relying on fixed infrastructure, enabling continuous exchange of motion data.
- Large-scale traffic scenario simulation: Generating diverse, controlled driving scenarios - both real and

synthetic - to train and test the decision-making process under variable conditions.

- AI-driven decision-making: Employing Reinforcement Learning (RL) to train an AI agent that evaluates dynamic traffic states and issues predictive braking or warning signals based on evolving conditions.
- Real-time integration and validation: Deploying and evaluating the trained system in real driving environments to assess its capacity for timely, reliable hazard detection and collision prevention.

C. The RoadSense Contribution

RoadSense advances the state of V2V hazard detection by implementing a complete closed loop, from multi-hop communication through learned hazard assessment to physics-based ego-car response, rather than stopping at packet-level metrics or scripted triggers. It introduces a four-layer realism framework that combines calibrated sensor noise, mesh-network impairments, environmental factors, and ego-car dynamics. This is supported by a hybrid dataset that combines real V2V logs with SMARTS (Scalable Multi-Agent Reinforcement Learning Training School for Autonomous Driving) based, domain-randomized simulations. The resulting system learns graded braking and warning policies using reinforcement learning and demonstrates low-cost deployability on a reproducible ESP32 (Espressif 32-bit Microcontroller) + GPS (Global Positioning System) + IMU (Inertial Measurement Unit) hardware platform, thereby linking realistic simulation to practical implementation.

D. The Structure of the Article

While *RoadSense* ultimately aims for a fully operational in-vehicle system, this paper emphasizes the development phase, during which the AI-based process is trained to recognize hazards and determine appropriate responses.

The article is structured as follows. It begins by describing the motivation for V2V-based hazard detection and reviews prior communication-centric and decision-centric research, establishing the gaps that *RoadSense* addresses. The system architecture is then presented, covering the mesh communication layer, the data-processing and filtering pipeline, the state-vector formulation, and the AI-based decision and action modules. The methodology section describes the development process, including the integration of prior knowledge from real V2V recordings, large-scale scenario simulation, and training of a reinforcement learning model, followed by a comparative analysis demonstrating improvements over existing approaches. The implementation and evaluation results are then presented for both simulated and real-world conditions. The paper concludes with a summary of contributions and potential future extensions.

II. RELATED WORK AND COMPARATIVE CONTEXT

Early research on V2V communication for driving safety focused primarily on demonstrating the feasibility and benefits of inter-vehicle data exchange for collision avoidance.

Yang et al. [2] introduced one of the first congestion-aware V2V protocols for Cooperative Collision Warning (CCW), demonstrating that low-latency, differentiated message delivery could significantly reduce reaction times during emergency braking. Later systems, such as COVCRAV [3], extended this concept to cooperative hazard signaling, introducing user-reported road hazards via interactive driver interfaces. Other simulation-based studies, such as Xie et al. [4], have demonstrated that V2V-augmented braking systems can outperform radar- or camera-based Automatic Emergency Braking (AEB) systems, particularly in limited-visibility conditions. Similarly, Joerer et al. [5], used the Veins simulator to show that inter-vehicle communication substantially reduces intersection collisions compared to perception-based control.

While these works established the communication benefits of V2V safety systems, they primarily relied on scripted hazard events and predefined control logic, rather than learning from contextual driving data. Over the past two decades, vehicle-safety research has advanced along two main directions: **communication-centric** and **decision-centric** approaches.

Communication-centric studies: typically based on the SUMO: Simulation of Urban MObility (SUMO) - Vehicles in Network Simulation (Veins) - Objective Modular Network Testbed in C++ (OMNeT++) (SUMO–Veins–OMNeT++) simulation stack, focus on optimizing network performance metrics such as Packet Reception Ratio (PRR), latency, jitter, channel load, and penetration rate. In these works, "hazards" are typically modeled as predefined events with scripted vehicle responses, thereby avoiding the deeper question of how a system should identify or evaluate hazards in context. Within this stack, SUMO models microscopic traffic mobility [6], [7]; OMNeT++ simulates the communication layer (using INET or Simu5G) [8]; and Veins synchronizes traffic and communication states [9], [10]. Together, they enable rigorous evaluation of V2V/V2X protocols such as IEEE 802.11p/ITS-G5 and C-V2X/NR-V2X under realistic traffic motion, leaving aside learned hazard assessment and vehicle-level control.

In contrast, **decision-centric studies** - most commonly using CARLA [11], [12] or SMARTS [13], [14] - train Reinforcement-Learning (RL) or autonomous-driving agents to make braking, yielding, or lane-change decisions in physics-based environments. However, these frameworks typically assume idealized communication conditions or omit V2V messaging altogether, leaving the learned policies unexposed to realistic network effects such as delay, loss, duplication, or out-of-order delivery.

More recently, **co-simulation efforts** have begun to bridge mobility dynamics and V2X networking—for example, MOSAIC with CARLA and SUMO with OMNeT++ setups [15], [16] coupled with physics engines or brake-control hardware. These studies achieve real-time, closed-loop integration in which communication signals influence vehicle dynamics. However, most still compromise along one axis: some employ learned decision-making under simplified communications, while others model realistic V2V impairments but rely on hand-crafted hazard logic (e.g., rule-based triggers, Time-To-Collision (TTC) thresholds). Consequently, few studies train and evaluate learned hazard-assessment policies under entirely realistic conditions that simultaneously incorporate calibrated sensor noise, network impairments, environmental effects, and physics-accurate ego-car dynamics.

Despite significant progress in V2V communication research, most prior studies have focused either on network-level metrics (latency, packet delivery, or congestion) or on scripted decision logic. Few have addressed the whole pipeline from communication through learned hazard assessment to physically accurate vehicle responses. The *RoadSense* project directly addresses this gap and distinguishes itself from prior research on V2V safety systems by integrating multiple domains of communication, AI decision-making, and physical realism into a unified, functional system within a reproducible, low-cost platform.

RoadSense differs from prior work along the following issues:

- Closed loop (comms → decision → action): *RoadSense* extends beyond packet-level metrics and scripted hazard triggers by *learning* hazard assessment via RL and linking it directly to a physics-based ego-response model.
- Four-layer realism: Unified modeling of sensor noise, mesh-network impairments, environmental conditions, and ego dynamics, all calibrated using an ESP32 + GPS + IMU testbed - rarely achieved concurrently.
- Hybrid data + domain randomization: Combines real V2V logs with SMARTS-based simulations to narrow the sim-to-real gap, providing a stronger rationale than simulation-only or proprietary crash corpora.
- Learned hazard assessment: Reinforcement learning (Deep Q-Network (DQN) → Proximal Policy Optimization (PPO)) maps the ego-car's plus neighboring vehicle's state vectors to graded braking and warning actions integrated in the complete closed loop - a capability usually explored in isolation elsewhere.
- Graded action policy: Beyond binary alert/no-alert logic, the learned policy maps confidence to graded braking and driver visualization or actuation, enabling interpretable, adaptive responses.
- Low-cost deployability: Implements a reproducible hardware reference design (ESP32 microcontroller, NEO-6M GPS, Motion Processing Unit model MPU-6050) demonstrating real-world feasibility rather than remaining a purely software-stack concept.

III. SYSTEM USAGE IN THE DEPLOYMENT PHASE

The operational flow in the final system within the car, once communication processes and AI-based decision-making have been trained and deployed, is shown in Figure **2**:

- Mesh communications: An ad-hoc V2V network that provides the vehicle with data from nearby cars, including those beyond line of sight (e.g., hidden by curves, trucks, or weather).
- Data collection and transformation: Gathers and merges data from surrounding vehicles with the ego-car's sensor data, removes duplicates and irrelevant information, and converts it into state vectors for an AI-based decision-making process.
- Decision-making: The trained AI decision process analyzes both local and received data to identify potential hazards ahead based on extensive reinforcement learning on a combination of real-world and simulated driving data.
- Action-taking: Based on the AI's assessment, this process determines the appropriate response - specifically, whether braking is necessary and, if so, the recommended braking intensity.
- Warning: Presents timely input to the braking mechanism or visual or auditory cues to the driver via the vehicle's dashboard interface.

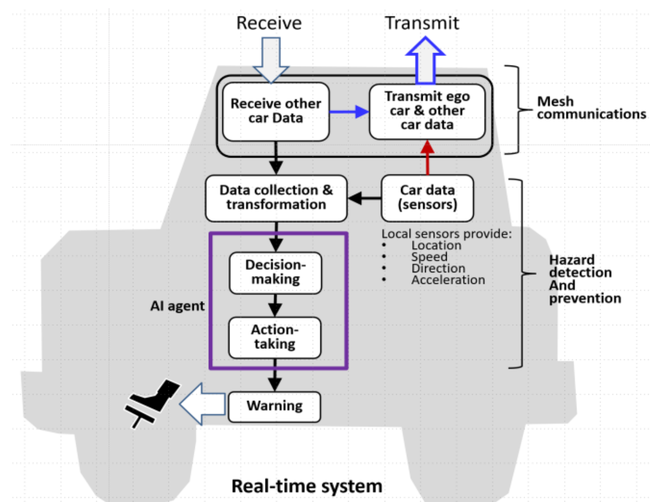

Figure 2. Architecture of the RoadSense real-time system: Sensor and V2V data are joined to support AI-based decision-making on braking commands.

A. *Mesh (Multi-Hop) V2V Communication Framework*

Each vehicle transmits its local sensor data to surrounding vehicles. To avoid communication congestion, as described by Xue Yang et al. [2], the transmission size was kept to a bare minimum, including only GPS location, acceleration, and identification. Nearby vehicles retransmit both their own data and the data they receive (and filter) to others in their vicinity, as shown in Figure 3. This creates an ad hoc, extensible mesh network in which messages are forwarded hop-by-hop across multiple vehicles. Such a multi-hop communication scheme significantly extends the effective range of data exchange, allowing each car to receive information from vehicles well beyond its direct line of sight and beyond the direct transmission distance.

For example, a car 1 km ahead can send data to the ego-car via intervening vehicles, even though it is beyond the ego-car's direct range. The mesh's redundancy ensures that if one path fails, the message can still be delivered via alternative routes, improving reliability.

B. Data Collection and Transformation Process

Upon receiving incoming data, the system standardizes and validates raw inputs from surrounding vehicles, removing duplicate, irrelevant, and incomplete information to ensure high-quality data for decision-making. The data is then time-aligned and transformed into the ego-car frame. In addition, the system performs consistency checks on GPS coordinates, timestamps, and sensor reliability to ensure that only verified, accurate data are processed and used.

Figure 3. Mesh V2V communications: Vehicles broadcast data to nearby cars, which rebroadcast it with their own data, forming a multi-hop mesh that propagates messages.

1) Cone-based filtering

Before entering the AI decision-making process, all incoming data from the mesh network undergoes filtering to retain only data relevant to the AI predictions. Each vehicle transmits its acceleration and GPS location. For hazard detection, only vehicles traveling ahead in the same lane and direction within a specified distance are considered relevant, thereby defining a cone-shaped region of interest extending forward from the ego-car, as shown in

Figure 4. The AI component analyzes the motion vectors of nearby vehicles within this region to assess potential collision risks and congestion trends. By discarding irrelevant, redundant, duplicate, and distant data, the filtering process ensures that only contextually significant information is retained for trajectory analysis and decision-making, thereby allowing each vehicle to operate efficiently without overloading the prediction algorithm. In addition, to further reduce unnecessary communication congestion, only messages that passed the filtering are retransmitted.

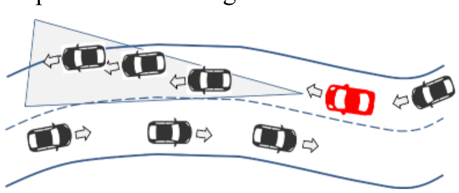

Figure 4. Transmitted car data filters out irrelevant cars by focusing on a cone-shaped area in the direction of travel.

2) State Vector Generation

Input into the decision-making process is provided by a state vector that represents the ego-car's state and the states of surrounding vehicles. The state vector includes:

- Ego-car parameters: speed, acceleration, azimuth, lane index/offset, and position in the ego-centric frame. All calculated locally from consecutive GPS locations and an accelerometer sensor.
- Traffic-context parameters (up to N neighbors): for each neighboring vehicle, local calculations generate relative distance and bearing, relative velocity and acceleration, with respect to the ego-car. This context encodes how the ego-car is positioned and moving relative to others; for example, whether a vehicle is rapidly decelerating sharply ahead, and how fast and how close the ego-car is to others ahead.
- Auxiliary/environmental parameters (optional): speed limit, curvature, road friction estimate, weather/visibility flags, given that they are available.

C. Decision-making Process

This is the AI-based process resulting from extensive reinforcement learning on real and simulated driving data, which will be explained in detail in the development phase section.

D. Action-taking Process

The *Action-Taking* process interprets the raw output of the decision-making process and converts it into human-readable or system-level commands. It does not generate its own predictions; instead, it maps the AI's continuous confidence values into discrete, safety-oriented actions and alert categories. Rule-based logic derived from kinematic safety models governs this mapping:

- 0.0–0.3 → No alert (safe distance maintained)
- 0.3–0.6 → Caution alert ("Light Brake Needed")
- 0.6–1.0 → Emergency alert ("Immediate Brake Required")

These thresholds can be dynamically adjusted based on contextual factors such as road friction, speed limits, predicted time-to-collision, and weather (Figure 5). The decision logic may also select the appropriate alert modality (visual, audio, or haptic) and verify that braking recommendations remain stable across consecutive frames before activation, ensuring both reliability and driver trust.

E. Warning Process

The *RoadSense* project defines warnings (or alerts) as the core output of the system's hazard-detection logic. Their role is to:

- Warn drivers of imminent risks such as potential collisions, sudden braking, or unsafe speeds.
- Provide graded or prioritized notifications, distinguishing critical, cautionary, and convenience-level messages.
- Improve reaction time by informing drivers before hazards are visible to sensors or the human eye.

The alerts are intended to enhance situational awareness rather than replace driver control; the system serves as a driver-assistance layer rather than an autonomous intervention mechanism.

IV. AI Training in the Development Phase

The ego-car's decision-making process relies on an AI model trained to accurately predict brake-warning signals by analyzing both local sensor inputs and remote data streams received from upstream vehicles via a mesh network. To effectively train this model, a simulation environment is constructed to expose it to a wide range of driving scenarios, thereby enabling robust learning of collision-avoidance behaviors.

Training involves generating data for both the ego-car and the surrounding vehicles, as shown in Figure 5. Moreover, the ego-car's physical dynamics must be simulated in real time to generate data and simulate the car's reaction whenever the decision-making process initiates a braking action.

The primary focus of the training phase is on the following key processes shown in Figure 5:

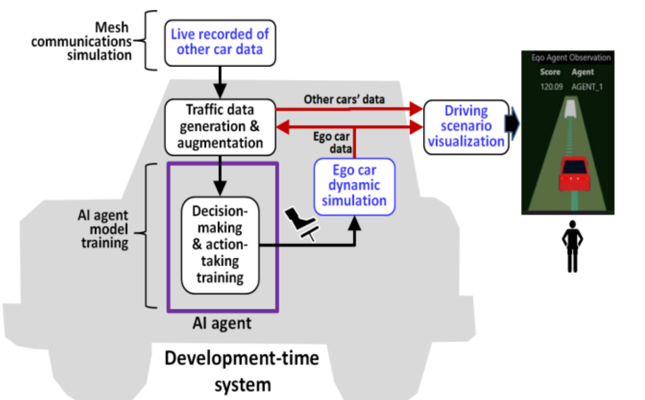

Figure 5. In the *RoadSense* simulation framework, recorded and synthetic data are used to train the decision-making AI.

- **Traffic Scenarios Data Recording and Augmentation**: Actual message communications are recorded in vehicles deployed with sensors and used to generate everyday traffic scenarios for simulation. Because real crash data are rare, simulated crashes and variations of traffic scenarios are generated to enrich the simulator dataset.
- **AI Agent Model Training**: The real and simulated driving-generated data is fed into the AI model that is trained using reinforcement learning to detect and respond to hazards.
- Ego-car Dynamics Simulation: In the simulator, a physical model of the ego-car and its dynamic properties is employed to replicate real driving behavior and calculate the vehicle's physical response to the braking actions recommended by the decision-making process.
- The simulation serves two primary purposes:
  - **Physics-Based Scenario Replay**: It integrates simulated data into the data collection pipeline, providing accurate representations of the ego-car's physical state. This enables realistic progression through driving scenarios by combining recorded or generated data from surrounding vehicles with physics-based modeling of the ego-car's behavior during critical phases, such as braking.
  - **AI Model Evaluation and Training**: It facilitates assessment of the decision-making process by determining whether the AI avoids collisions and maintains safe driving conditions. This feedback loop supports reinforcement learning, allowing the AI model to improve its control strategy based on the simulated outcomes of each decision step.
- **Driving Scenario Visualization**: During development, a visualization module was implemented to display real-time vehicle positions and trajectories following braking or warning actions initiated by the decision-making system. This visualization enables assessment of driving outcomes and potential collision scenarios, providing valuable feedback on the system's effectiveness and response accuracy.

The above processes and their design are described in greater detail in the following sub-sections.

A. *The Simulation Environment and Data Generation*

The simulator is designed to utilize data that closely reflects real-world driving conditions. Vehicles equipped with dedicated hardware record GPS coordinates, accelerometer readings, and timestamps while operating on actual roads. These recordings capture diverse scenarios, including approaching intersections, navigating curves, and driving in low-visibility conditions. Additionally, the vehicles implement mesh communication protocols and log transmissions received from nearby cars. This collected data serves as the foundation for generating realistic simulation scenarios that include trajectories and mesh communications as expected in the real world.

In addition to collecting real-world data, synthetic data is programmatically generated to support various testing and training objectives. This includes increasing traffic density through data replication, simulating sudden braking events and full-stop collisions, and introducing communication anomalies such as increased latency, message loss, and network congestion.

The ego-car's motion is simulated using its physical parameters and kinematic equations to model its trajectory before and after any braking command issued by the decision-making module. Initial path and velocity vectors are specified as input parameters and are dynamically updated based on the AI-driven decision process's evolving outputs.

Traffic scenarios are implemented in SMARTS, an open-source simulator for multi-agent reinforcement learning [13] extended with our own communication and noise-modeling modules. Instead of relying on proprietary accident datasets, real-world constraints are embedded directly into the simulation. The scenario scripts:

- Select a road segment (urban arterial or multi-lane highway) and legal-speed geometry.
- Sample traffic density and vehicle type distribution.

- Initialize actors with lane, speed, headway, and position relative to the ego-car.
- Trigger events (e.g., hard breaking, cut-ins, stop-and-go waves, obscured obstacles) in randomized time windows.

To promote generalization, we apply domain randomization within realistic bounds informed by published driving statistics and our hardware measurements. Parameters varied include vehicle speeds (±10–20%), packet loss (0–30%), message-delay jitter, GPS noise at the meter scale, and standard lane/vehicle dimensions.

The resulting environment exposes the AI agent to diverse hazard scenarios, such as rapid deceleration, high-density traffic, and occluded obstacles. It enables safe learning from situations that cannot be recreated in real vehicles. Simulated and real V2V data from our testbed are combined to form a comprehensive training set, thereby improving accuracy and robustness prior to deployment.

B. Deep Q-Learning and PPO-Based Adaptive Decision-Making

We implemented a Reinforcement Learning (RL) model in *RoadSense* to issue timely braking recommendations to avoid collisions.

The RL objective rewards correct hazard responses (timely slow/stop to avoid collisions) and penalizes failures (late response, collisions, unnecessary harsh braking).

Two reinforcement learning algorithms were employed:

1) Deep Q-Learning (DQN): used during early development to train the agent on discrete binary actions (e.g., brake / no brake).

2) Proximal Policy Optimization (PPO): adopted for the final system due to its superior performance in continuous control domains and its robustness against unstable policy updates, Schulman [17], Minih [15]. PPO constrains policy updates within a trust region, ensuring stability and preventing overfitting to transient simulation states. Simulation runs performing RL training for interactions between 3 cars: an ego-car and two cars in front, have been shown to successfully converge after several hundred-thousand Epochs, as shown in Figure 6.

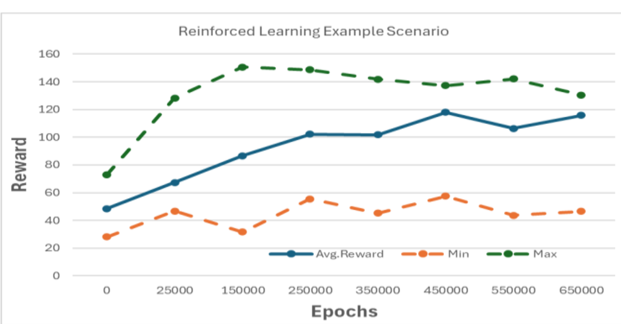

Figure 6. Example of a Reinforcement-Learning session that converged after 650K Epochs.

C. Driving Scenario Visualization and Warning

For visualization, a Unity-based environment is used to display vehicle positions and trajectories before and after braking actions, illustrating the vehicle's dynamic response to the decision-making system's braking or warning commands, as shown in

Figure 7. This enables the assessment of driving outcomes and potential collision scenarios, providing valuable feedback on the effectiveness and timing of the decision-making process.

V. ROADSENSE DEVELOPMENT, INSTALLATION AND TESTING

The initial phase involved assembling and validating multiple hardware boards, which were subsequently installed in vehicles and powered via the onboard USB interface. As part of the Mesh communications system, custom firmware was developed for the microcontroller to perform sensor monitoring, transmit telemetry data at 1 Hz, and operate as a node in an ad hoc mesh network. To reduce communication overhead, filtering mechanisms were implemented to selectively retransmit only data relevant to trailing vehicles, specifically, information originating from cars within the forward extended cone, as described earlier. Additionally, for development and debugging, the firmware was enhanced to monitor and log sensor readings and network communications.

The second phase involved capturing and recording real-world vehicular telemetry data across various driving scenarios, including deceleration before intersections and sharp turns. These raw recordings were subsequently processed and programmatically augmented to synthetically expand the dataset. This augmentation introduced variability and increased the diversity of traffic scenarios, resulting in a comprehensive database encompassing both typical driving patterns and artificially generated incident conditions.

In the third phase, the SMARTS simulation framework [13] was employed to emulate traffic conditions and mesh communications using the previously recorded dataset. Within the simulator, an ego-car was dynamically controlled using physical parameters to emulate realistic physical driving behavior. An integrated AI module was tasked with issuing predictive braking alerts, enabling the ego-car to decelerate smoothly and avoid collisions with vehicles ahead under varying traffic conditions. The simulator iteratively executed each traffic scenario over a substantial number of epochs, during which the AI module was trained using reinforcement learning techniques to optimize its decision-making policy.

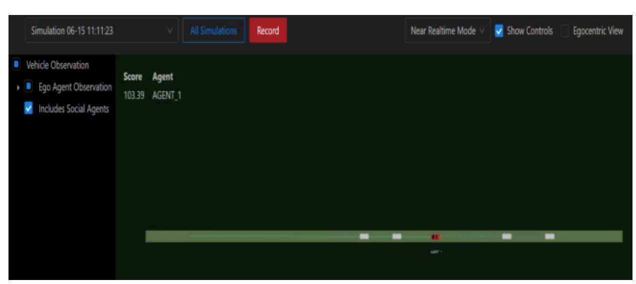

Figure 7. Visualization of the response of the ego-car to the warnings issued by the decision-making processes.

 ISBN: 978-1-68558-348-4

Initial validation of the trained model was conducted using an external visualization process built on the Unity engine. After the learning algorithm converged, this component was used to render graphical representations of each scenario's outcome, enabling qualitative assessment of the AI module's performance.

The final phase involved deploying the trained AI agent onto the ego-car's onboard hardware platform and conducting field tests under real-world driving conditions. Validation procedures focused on verifying the timing and accuracy of the brake alert relative to the human driver's decisions. The results demonstrated that, even when the driver's visibility of preceding vehicles was obstructed, the AI agent reliably initiated braking promptly, effectively mitigating collision risk.

VI. Results and Discussion

The *RoadSense* system was implemented and validated through a staged process combining simulation and road testing. During the simulation phase, the Reinforcement-Learning (RL) agent was trained using a hybrid dataset comprising both real V2V mesh recordings and SMARTS-generated traffic scenarios. Domain randomization was applied to communication latency, packet loss, GPS jitter, and environmental factors. The model successfully converged within approximately 650,000 training rounds, yielding stable hazard detection and braking policies.

During controlled simulations, the trained AI consistently generated timely, graded alerts across diverse driving contexts—including sudden braking, stop-and-go waves, and reduced-visibility conditions, demonstrating accurate state-vector interpretation and effective mapping of policy confidence to braking intensity.

Field deployment using a low-cost ESP32 paired with GPS and IMU sensors confirmed that the end-to-end mesh–decision–action loop functioned in real time, achieving end-to-end latency below 2 seconds. Tests under partial occlusion (e.g., blocked line of sight by larger vehicles) showed that the AI module could initiate braking before the driver's visual recognition of hazards, confirming predictive behavior consistent with simulation results. The graded alert mechanism improved driver reaction timing and system interpretability without false or unnecessary interventions.

VII. Summary, Conclusions and Future Work

In summary, existing research either optimizes the communication layer in isolation or develops decision-making agents under idealized conditions, rarely integrating both within a realistic, sensor-calibrated, physics-accurate framework. *RoadSense* presents a comprehensive, closed-loop V2V hazard detection system that integrates realistic communication models, physics-driven vehicle dynamics, and reinforcement-learning-based decision-making algorithms within an accessible, cost-effective platform.

Unlike previous studies that focus on either network metrics or scripted control logic, *RoadSense* merges four calibrated realism layers—sensor noise, mesh-network impairments, environmental variation, and vehicle dynamics into a single end-to-end architecture that learns hazard semantics rather than applying fixed thresholds.

The hybrid training methodology, which combines real-world mesh-network data with domain-randomized SMARTS simulations, enabled the RL agent to adapt to noisy communications and unpredictable driving conditions. Results confirm that learned policies can operate effectively on embedded hardware, providing graded alerts that enhance situational awareness and reduce driver reaction times even beyond line of sight.

Subsequent development will expand *RoadSense* to encompass additional sensors, more sophisticated mesh communication protocols, multi-vehicle coordination, and traffic scenarios that extend to additional hazardous conditions.

References

[1] F. d. P. Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," Sensors, vol. 17, p. 271, 2017.

[2] X. Yang, L. Liu, N. H. Vaidya and F. Zhao, "A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning," in 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), Networking and Services, Cambridge, MA, USA, 2004.

[3] F. Outay, H. Bargaoui, A. Chemek, F. Kamoun and A Yasar, "The COVCRAV project: Architecture and design of a cooperative V2V crash avoidance system," in The 3rd International Workshop on Connected & Intelligent Mobility, Coimbra, Portugal, 2019.

[4] Z. Xie, D. Wu, D. Liu, H. Huang and B. Xie, "V2V Communication-based AEB Validation in Traffic Accident Simulation Scenario," in ICFEICT 2021: International Conference on Frontiers of Electronics, Information and Computation Technologies, Changsha, China, 2021.

[5] S. Joerer, B. Bloessl, M. Huber, A. Jamalipour and F. Dressler, "Assessing the impact of inter-vehicle communication protocols on road traffic safety," in Proceedings of the 6th annual workshop on Wireless of the students, by the students, for the students, Maui, Hawaii, USA, simmer.

[6] M. Behrisch, L. Bieker-Walz, J. Erdmann and D. Krajzewicz, "SUMO – Simulation of Urban MObility: An Overview," in SIMUL 2011, 2011.

[7] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO—Simulation of Urban MObility," International Journal on Advances in Systems and Measurements, vol. 5, no. 3, pp. 128-138, 2012.

[8] R. H. A. Varga, "An Overview of the OMNeT++ Simulation Environment," in Proc. of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SimuTools), 2008.

[9] C. Sommer, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," IEEE Transactions on Mobile Computing, vol. 10, no. 1, pp. 3-15, 2011.

[10] C. Sommer et al., "The open source vehicular network simulation framework," in Recent Advances in Network Simulation, Springer, 2019, pp. 215-252.

[11] D. Alexey, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in Conference on Robot Learning (CoRL), 2017.

[12] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, J. Araluce and L. Bergasa, "Reinforcement Learning-Based Autonomous Driving at Intersections in CARLA Simulator," Sensors, 2022.

[13] M. Zhou et al., "SMARTS: An Open-Source Scalable Multi-Agent RL Training School for Autonomous Driving," in Proceedings of the 2020 Conference on Robot Learning, 2021.

[14] A. Rasouli, "Driving SMARTS Competition at NeurIPS 2022: Insights and Outcome," in NeurIPS 2022 Competitions Track, 2022.

[15] D. Cui, Y. Shen, H. Yang, Z. Huang, K. Rush, P. Huang and P. Bujanovic, "Extensible Co-Simulation Framework for Supporting Cooperative Driving Automation Research," Transportation Research Record, Journal of the Transportation Research Board, pp. 1067-1079, 2022.

[16] A. Marchetta, A. Coppola, M. Cinque, M. Fiorentino and G. N. Bifulco, "An Eclipse MOSAIC-Based Hardware-in-Loop V2X Co-Simulation Framework for CCAM Services," in IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 2023.

[17] S. Zeadally, J. Guerrero and J. Contreras, "A tutorial survey on Vehicle-to-Vehicle communications," Telecommunication Systems, vol. 73, no. 3, pp. 469-489, 2020.

[18] K. N. Toshiyuki Hagiya, "Acceptability evaluation of inter-driver interaction system via a driving agent using Vehicle-to-Vehicle communication," in The 11th Augmented Human International Conference, Winnipegm Manitobam Canada, 2020.

[19] S. Banerjee et al., "Link-Adaptation for Improved Quality-of-Service in V2V Communication using Reinforcement Learning," in The Second International Conference on AI-ML Systems, Bangalore, India, 2023.

[20] Y. Song, H. Chunxi, X. Weiyin and H. Dengbo, "HMI Design for Chain-Braking Event Based on V2V Communication," in AutomotiveUI '23 Adjunct: Adjunct Proceedings of the 15th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Ingolstadt, Germany, 2023.

[21] X. Zhang, J. Li, J. Zhou et al., "Vehicle-to-Everything Communication in Intelligent Connected Vehicles: A Survey and Taxonomy," Automotive Innovation, vol. 8, p. 13–45, 2025.

[22] S. Lu, "Modeling dynamics of traffic flow, information creation and spread through Vehicle-to-Vehicle communications: A kinetic approach," International Journal of Non-Linear Mechanics, vol. 175, 2025.

[23] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," vol. 99, no. 7, p. 1162–1182, 2011.

[24] ETSI, "Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band, ETSI EN 303 797 V2.1.1," 2019. [Online]. Available: https://standards.globalspec.com/std/10146015/en-302-571.

[25] S. I. T. Standard, " Dedicated Short Range Communications (DSRC) Message Set Dictionary," 2015. [Online]. Available: https://doi.org/10.4271/J2735_201509.

[26] G. Kahn, A. Villaflor, Vitchyr Pong, Pieter Abbeel, Sergey Levine, "Uncertainty-aware reinforcement learning for collision avoidance," arXiv preprint, arXiv:1702.01182, 2017.

[27] L. Cao et al., "Driving-policy adaptive safeguard for autonomous vehicles using reinforcement learning," arXiv preprint, arXiv:2012.01010, 2020.

[28] H. Chen, X. Cao, L. Guvenc and B. Aksun-Guvenc, "Deep-reinforcement-learning-based collision avoidance of autonomous driving system for vulnerable road user safety," Electronics, vol. 18, no. 10, pp. 1952, 2024.

[29] S.H. Ashwin and R. Naveen Raj, "Deep reinforcement learning for autonomous vehicles: lane keep and overtaking scenarios with collision avoidance," International Journal of Information Technology, vol. 15, no. 7, 2023.

[30] A. M. A. Behzadan, "Adversarial reinforcement learning framework for benchmarking collision avoidance mechanisms in autonomous vehicles," 2018. [Online]. Available: https://arxiv.org/abs/1806.01368.

[31] S. Zeadally et al., "A tutorial survey on Vehicle-to-Vehicle communications," Telecommunication Systems, vol. 73, no. 3, 2020.

[32] X. Yang, L. Liu, N. H. Vaidya and F. Zhao, "A Vehicle-to-Vehicle communication protocol for cooperative collision warning," in The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004, pp. 114-123.