IEEE Open Journal of
Intelligent Transportation Systems

Received 18 December 2024; revised 23 March 2025; accepted 30 August 2025. Date of publication 5 September 2025;
date of current version 17 September 2025.

Digital Object Identifier 10.1109/OJITS.2025.3606539

Self Healing of a Mixed Autonomy Traffic System Using Reinforcement Learning and Attention

SAFIA FATIMA [1,2], KAI OLAV ELLEFSEN [2], AND LEON MOONEN [1]

[1] Data-Driven Software Engineering Department, Simula Research Laboratory, 0164 Oslo, Norway
[2] Department of Informatics, University of Oslo, 0316 Oslo, Norway

CORRESPONDING AUTHOR: L. MOONEN (e-mail: leon.moonen@computer.org)

This work was supported by the Research Council of Norway through the cureIT Project under Grant 300461.

ABSTRACT As urban traffic becomes increasingly complex with the integration of connected and autonomous vehicles alongside human-driven vehicles, there is a critical need for adaptive traffic management systems capable of self-healing in response to disruptions. This paper introduces TS2RLA ("Traffic System Recovery using Reinforcement Learning and Attention"), a novel framework for self-healing in mixed-autonomy traffic systems by combining deep reinforcement learning with an attention mechanism to optimize traffic flow and recover from faults in various scenarios in a mixed-autonomy traffic environment. We evaluated TS2RLA in four complex traffic scenarios: bottleneck, figure-eight, grid, and merge. Our results demonstrate significant improvements over the baseline model, showing an average of 86.74% reduction in crashes, 71% improvement in speed and traffic throughput, and robust performance under diverse and complex traffic conditions. Moreover, our experiments show that TS2RLA leads to a significant reduction in CO2 emissions and fuel consumption. TS2RLA's attention-based approach shows particular benefits in bottleneck and figure-eight scenarios, demonstrating its ability to adapt to complex, multi-factor traffic situations. For scenarios that TS2RLA had not been trained on before, it performs even more favorably than the baseline, with a 96.8% crash reduction and 95.3% throughput improvement. This shows its ability to adapt effectively to new traffic conditions. Overall, we conclude that TS2RLA could significantly improve the safety, efficiency, and capacity of real-world traffic systems, particularly in dynamic urban environments. As such, our work contributes to the field of intelligent transportation systems by offering a versatile self-healing framework capable of managing the complexities of mixed-autonomy traffic.

INDEX TERMS Traffic system, mixed-autonomy, reinforcement learning, attention network, recovery, self-healing.

I. INTRODUCTION

SELF-ADAPTIVE systems (SASs) are dynamic systems that respond to environmental demands. The primary goal of SASs is to improve system performance, reliability, and resilience in the face of changing conditions [1]. Due to their dynamic nature and efficiency in adapting to work in uncertain scenarios, the self-adaptive system has gained importance in modern software engineering, particularly in complex systems. Such systems include autonomous systems, smart grids, LTE communication networks, traffic systems, Internet of Things (IoT)-based systems, and the like [2], [3], [4]. The term SASs is a combination of self-protection, self-healing, self-optimization, and self-configuration [5] where each property has its importance. Self-healing is a key application of self-adaptive systems, which indicates that a system can monitor and correct faults. Self-healing techniques vary, with some using data-driven approaches and others using model-based methods. Data-driven approaches utilize historical and real-time data to detect faults and provide appropriate recovery actions. In contrast, model-based techniques rely on predefined

The review of this article was arranged by Associate Editor Johannes Betz.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

system models, including components, interactions, and expected behaviors [6], [7], [8]. To achieve self-healing, many methods have been utilized [9], [10], [11], including artificial immune systems [12], constraint-based programming [13], conventional machine learning [14], [15], and neural networks [16], with recent studies exploring deep reinforcement learning [17], [18] to achieve self-healing capacity in a software system, but there is still much more to address to make a fully autonomous general self-healing framework that applies to a wide range of self-adaptive systems [19]. Today, traffic systems are facing a "mixed-autonomy" scenario, where both autonomous and non-autonomous vehicles share the road. In recent years, much work has been focused on managing traffic flow and signals [20], [21], [22], but recovery of mixed-autonomy scenarios has not been discussed to our knowledge. Recovery of these systems is crucial due to their complex and dynamic nature. This mixed environment significantly increases the complexity of traffic management due to factors such as diverse vehicle behaviors, complexity of interaction, and safety considerations. In addition, not much work has been done on handling large action sequences to manage these complex systems.

To address these challenges, we study the traffic system in the context of self-healing and propose a novel self-healing framework called *TS2RLA ("Traffic System Recovery using Reinforcement Learning and Attention")* to recover the system when faults occur. Our framework uses an attention-based reinforcement learning (RL) policy to restore traffic systems to their normal state. The subject system for the proposed framework is a mixed-autonomy traffic system, consisting of connected autonomous vehicles (CAVs) and human-driven vehicles. We designed this framework to handle large sequences of high-dimensional data from multiple traffic scenarios. To empirically evaluate our framework for real-world application, we apply it to four specific scenarios commonly encountered in traffic systems: Bottleneck, Merge, FigureEight, and Grid. These scenarios were carefully selected to represent a wide range of typical traffic challenges in urban and highway environments. We use RL as a continuous learning approach that observes the subject system but only activates when a fault is detected and recovery is needed. Additionally, we integrate attention mechanisms with RL to handle the large sequence of inputs and variable dimensions across different scenarios. This attention mechanism improves the recovery framework by focusing on the relevant parts of the subject system. RL with an attention layer can lead to more efficient and effective self-healing, especially for our wide range of traffic scenarios. Our attention-based RL policy reduces unnecessary computation and improves overall efficiency by focusing only on relevant information. Our framework is optimized and flexible because it only activates after fault detection.

The main contributions of this paper are:

1) Introducing a way to study traffic systems from a self-healing perspective, providing insights into how a traffic system can autonomously detect, respond to, and recover from disruptions in a mixed-autonomy environment while simultaneously optimizing traffic flow.
2) A novel recovery approach, TS2RLA, for mixed-autonomy traffic environments using reinforcement learning (RL) and an attention network. RL detects and recovers from faults, while the attention mechanism handles large observation sequences.
3) We empirically evaluate our work in four scenarios (FigureEight, merge, bottleneck, and grid) and show significant results compared to a baseline (RL-based recovery policy without the attention mechanism).
4) We provide a replication package with code and data for our study, to enable replication and verification of our results, and allow others to build on our work [23].

The paper is structured as follows. Section II provides a brief background. Section III explains the proposed approach, followed by an empirical evaluation in Section IV. Section V discusses the results and main findings. We then present a brief section on threats to validity in Section VI. Section VII reviews related work. The paper concludes in Section VIII with a summary of our contributions and future research directions.

II. BACKGROUND

A. SELF-HEALING SYSTEMS

Self-healing is one of the most important characteristics of SASs. It ensures that SASs can recover from faulty or abnormal states. In the last decade, extensive research has focused on integrating self-healing in various fields including the IoT, cyber-physical systems, energy management systems, communication, robotics, and many more [24], [25], [26], [27], [28].

Although self-healing has been utilized in many areas, less work has been done to integrate it into traffic systems. Some work on self-organization, which is close to our work, has been done [29]. In this paper, the authors manipulate traffic lights based on road capacity. However, this work focuses on organizing the traffic flow rather than recovering or self-healing the entire traffic system.

B. REINFORCEMENT LEARNING

Reinforcement Learning is usually formulated as a Markov Decision Process (MDP), a framework that enables agents to learn optimal behaviors in a given environment. The core objective is to maximize cumulative rewards through environmental interactions. An MDP is defined by $M(S, A, P, R, \gamma)$, where S is the state space, A is the action space (discrete or continuous), $R : S \times A \mapsto \mathbb{R}$ is the reward function, γ is the discount factor for future rewards, and $P : S \times A \mapsto S$ is the transition probability. The Markov property, a key aspect of this framework, dictates that decisions are based solely on the current state, eliminating the need for historical data. This property is crucial for RL control decisions, facilitating efficient and focused learning. The MDP framework provides a structured approach to problem solving in RL, allowing agents to navigate complex environments and make decisions based on immediate states and potential future rewards. In our work, we chose a Deep Reinforcement Learning (DRL) algorithm to train CAVs in the simulated environment. DRL has significantly advanced the field of RL in recent years. DRL leverages deep neural networks to solve the MDP model by learning the weight parameters θ through environmental exploration. These models incorporate a policy π, which determines actions based on states $\pi(a|s)$, and a value function $v_\pi s$ that estimates maximum rewards for the current policy $s \epsilon S$. Unlike traditional RL, which struggles with high-dimensional state spaces, DRL employs deep neural networks to approximate functions over extensive input and action state spaces, enabling the solution of complex problems. The policy and value functions in DRL are learned through these deep net models to predict future actions.

Next, we present a concise overview of the DRL model used for training and validating CAVs in simulated mix autonomy traffic system scenarios. We selected the model based on two criteria: its popularity within the DRL-based Autonomous Driving (AD) research community and its ability to handle both discrete and continuous action spaces, enabling multi-agent testing.

Proximal Policy Optimization (PPO): is a DRL algorithm that extends policy gradient (PG) methods [30]. While vanilla PG suffers from high gradient variance, PPO addresses this issue by introducing constraints such as a clipped surrogate objective and a penalty coefficient called KL. These improvements have made PPO a popular choice in the field of DRL in recent years. Also, recent studies have shown that PPO has demonstrated more promising results in the field of traffic signal control, navigation of autonomous vehicle and traffic management as compared to other algorithm [31], [32], [33].

C. ATTENTION NETWORK

Attention mechanisms have changed the traditional way of deep learning. By helping the model focus on important parts of the input sequences, it greatly improves how neural networks handle sequential data [34]. The core concept is represented as A(Q, K, V), where Q, K, and V denote queries, keys, and values. The fundamental attention function is expressed as follows.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Here, d_k is how large the keys are and helps prevent very small numbers from occurring in softmax operation.

Self-attention is really important in modern systems. It enables a sequence to analyze its own internal relationships, allowing different elements within the sequence to interact and inform each other. For an input sequence X, the model learns three different ways to change it, making the Q, K, and V matrices: $Q = XW_Q$ $K = XW_K$ $V = XW_V$ where W_Q, W_K, and W_V are matrices that the model learns.

Positional encoding, denoted as $\text{PE}(\text{pos}, 2i)$, is crucial in attention mechanisms, incorporating essential sequential order information that the attention operation inherently lacks. By embedding positional data into input representations, models can better understand the context and relationships between elements. In practice, attention mechanisms offer significant advantages over traditional sequence processing methods. They enable parallel computation, improving efficiency by simultaneously processing multiple sequence elements. Unlike recurrent architectures, attention can directly model relationships between distant elements, capturing long-range dependencies without sequential processing limitations. The attention weights provide valuable insights into the model's decision-making process, enhancing interpretability. In addition, attention mechanisms are highly flexible and adaptable to various types of input data and tasks, making them versatile tools in machine learning and AI applications. These characteristics have led to the widespread adoption and success of attention mechanisms in diverse deep learning domains. Researchers have come up with new versions of the basic attention mechanism. These include mixes of local and global attention [35], cross-attention for systems with separate encoding and decoding parts [36], and new ways to encode positions [37]. These mechanisms are now key parts of many different systems, from those that work with language to those that work with images. They are really good at understanding complex relationships in input data.

III. PROPOSED APPROACH

In this section, we will explain the proposed framework TS2RLA for the self-healing of the traffic system and its operational flow. The framework consists of two main components: *Environment*, which includes the scenario of the traffic system and information on fault interference, and *Agent*, which includes a recovery framework and action plans for self-healing, as shown in Figure 1. The operational flow will detail the procedural steps and algorithms for initializing the environment, deploying agents, executing traffic scenarios, and analyzing results. This comprehensive approach aims to systematically recover mixed-autonomy traffic systems once the fault has been detected.

A. ENVIRONMENT

The environment consists of two block Mixed-autonomy Traffic System where we have shown the traffic scenarios and Fault Interference where the fault is detected as shown in Figure 1.

1) MIXED-AUTONOMY TRAFFIC SYSTEM

The Mixed-autonomy Traffic System closely mirrors real-life situations where CAVs and human-driven cars coexist. This system is designed to study the interaction between

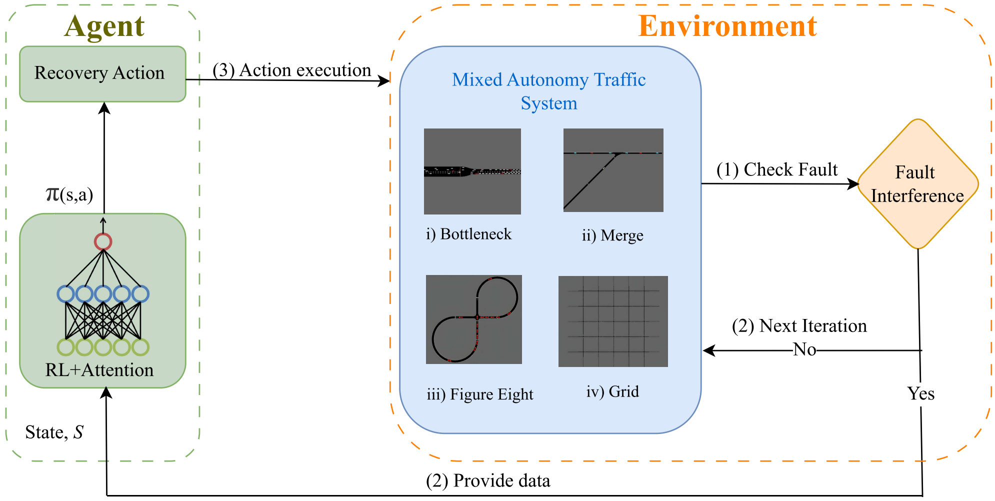

FIGURE 1. Proposed Framework for the Recovery of Traffic System. The agent on the left uses RL and attention network to recover the environment on the right side when one of the four driving scenarios faces a failure state.

RL-driven vehicles and those controlled by human drivers within the same traffic environment. In our proposed approach, we use Flow [38] to simulate the environment. Our approach includes four distinct scenarios in the environment. They are constructed to provide comprehensive information on the dynamics of mixed-autonomy traffic. These scenarios, as shown in Figure 1 in the block of *"Driving Scenarios"*, are based on previously established benchmarks for the Mixed-autonomy Traffic System [39]. Each scenario is crafted to highlight the theoretical and practical factors such as the interaction between RL algorithms and human decision-making, the efficiency of traffic flow, and the safety measures required to prevent collisions while maintaining manageable computational demands. By incorporating these scenarios, our approach aims to create a robust framework to evaluate the performance and effectiveness of RL-driven vehicles in real-world traffic conditions.

The four driving scenarios used while proposing our approach are shown in Figure 2.

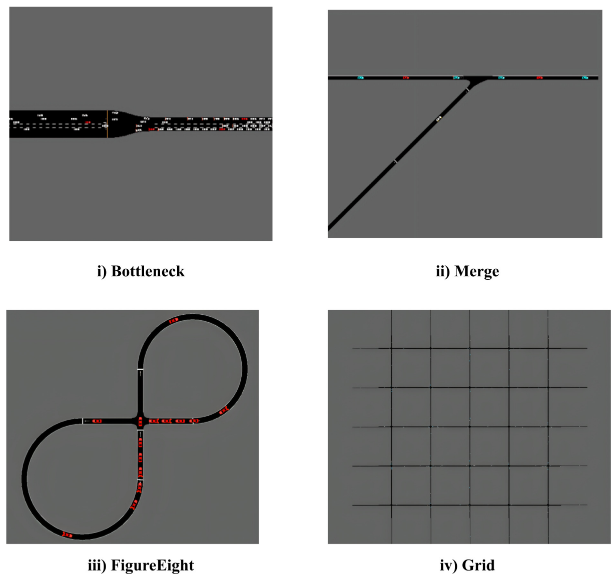

FIGURE 2. Driving Scenarios used for implementing TS2RLA framework.

1) Bottleneck: This scenario in our model is based on the Oakland-San Francisco Bay Bridge, where 16 non-high occupancy vehicle (HOV) lanes are reduced to eight and then to five. In our simulation environment, lanes decrease from 4N to 2N to N, N being a scaling factor. This setup illustrates the capacity drop phenomenon, where vehicle throughput decreases significantly once inflow exceeds a critical threshold. This effect leads to inefficiency in highway traffic.
2) Merge: This network shows how disturbances from vehicles merging onto a highway cause stop-and-go waves, reducing vehicle throughput, known as convective instability. In mixed-autonomy settings, some highway vehicles use local information to mitigate these waves. The open network allows the number of connected and autonomous vehicles (CAVs) to vary at any time.
3) FigureEight: This network simulates an intersection. In this setup with 14 vehicles, queues form as vehicles converge at the intersection and slow down to follow the right-of-way rules, leading to a significant decrease in average speed. In a mixed-autonomy scenario, some vehicles are designated as CAVs to manage the flow through the intersection and improve the overall speed of traffic.
4) Grid: This scenario represents a city with a grid-like structure, such as Manhattan. This problem aims to address issues in traffic light coordination. The goal is to create new traffic light control schemes that minimize the average delay per vehicle and promote fairness. Vehicles enter at the corners of the grid and travel straight. Each intersection has a traffic light that transitions from green to yellow for two seconds before switching to red, ensuring safety.

2) FAULT INTERFERENCE

This block is to detect interference in the traffic system. As soon as the interference has been detected, the information is sent to the Agent to take the appropriate action. For this proposed approach, we focus on three types of interference, that is, *crash*, *vehicle speed*, and *performance*. More details of each interference are provided in Section III-C.2.

B. AGENT

As illustrated in Figure 1, this block comprises two main components, which are the recovery agent framework of TS2RLA and the recovery actions.

1) RECOVERY AGENT IN TS2RLA FRAMEWORK

Our TS2RLA framework in creating a self-healing system for mixed-autonomy traffic systems consists of an agent based on RL. The different components of the recovery agent are shown in Figure 3.

The TS2RLA framework consists of four main building blocks for a successful recovery policy in our TS2RLA framework: (1) DRL policy with attention mechanism, (2) input features from driving scenarios, (3) actions for recovery based on the driving scenario, and (4) the reward functions designed for each mixed-autonomy driving system.

1) DRL policy with attention mechanism: A novel contribution in this paper is to propose a DRL policy that also involves attention layers. The policy TS2RLA learns to restore the driving system to its normal state by taking control of the CAV agents, as well as the traffic light systems. Our RL agents use PPO as a policy gradient method to learn a recovery policy by encountering a simulated environment in each training episode. The PPO helps perform on-policy learning within simulation instead of a dataset (replay buffer) type of learning. It also helps to focus on policy updates with stability while learning a change in data distributions, as well as to address a large hyperparameter initializing space. The summary of the DRL architecture, including the input, hidden, and output layers, is shown in Figure 3. The input state $S \subset R$ of our DRL algorithm receives a full observation of the state of the driving scenario where a fault has occurred. The input can contain information ranging from the position of the cars to their velocities, and other information discussed in Section III-B.2. Such input data is passed through multiple layers as shown in Algorithm 1 before reaching the output layer for control commands. In the output layer, we have a discrete action space, which the recovery agent uses to make recovery actions. All discrete actions are explained in further detail in 3 on page 1205.
2) Input State Space: As mentioned in 1, the TS2RLA policy handles high-dimensional state input ranging from 141 to 915 dimensions, depending on the scenario. The attention mechanism helps focus on the most relevant aspects of the traffic state during recovery, enabling efficient processing of sequential traffic data while capturing dependencies.

Algorithm 1 Process Input With Attention for TS2RLA

Input: State state_input from traffic scenario
hidden ← DenseLayer(state_input)
memory ← StoreAndRetrieveMemory(hidden)
Attention Mechanism Processing
query ← hidden · W_Q
key ← memory · W_K
value ← memory · W_V
scores ← $\frac{\text{query}\cdot\text{key}^T}{\sqrt{d_k}}$
attention_weights ← softmax(scores)
context ← attention_weights · value
Skip Connection and Normalization
context ← LayerNorm(context + hidden)
Feed-Forward Processing
output ← FeedForward(context)
output ← LayerNorm(output + context)
return processed state output

a) Bottleneck: This scenario involves calculating the average positions x_h and velocities v_h of human drivers in each lane for every edge segment, as well as the average positions x_c and velocities v_c of CAVs in each segment. It also includes measuring the outflow of the system in vehicles per hour over the last 5 seconds. The input representation consist of approximately 141 to 281 dimension states.
b) Merge: The input state received while handling a merge driving scenario includes the speeds and bumper-to-bumper gaps of vehicles both in front of and behind the CAVs, as well as the speed of the CAVs themselves, denoted as

$$\left[s := \left(v_{i\_lead}, v_{i\_lag}, h_{i\_lag}, v_i\right) \in \mathbb{R}^{nRL}\right] \quad (2)$$

where v_{i_lead} is the speed of the leading vehicle, v_{i_lag} is the speed of the lagging vehicle, h_{i_lag} is bumper-to-bumper gap with the lagging vehicle, v_i is the speed of the CAV, and nRL is the total number of CAVs. The input representation consist of approximately 25 to 85 dimension states.
c) FigureEight: The state is represented by a vector that contains the velocities v_i and positions x_i of each vehicle in the network, arranged according to their positions. This vector is denoted as

$$\left[s := (v_i, x_i)_{i=0}^{k-1} \in \mathbb{R}^{2k}\right] \quad (3)$$

where k is the number of vehicles. The positions are defined relative to a predetermined starting point in each setting of a scenario. The input representation consist of approximately 339 to 915 dimension states.
d) Grid: The input state when dealing with a grid scenario is represented by a vector that includes the velocities v_i and positions x_i of each vehicle in

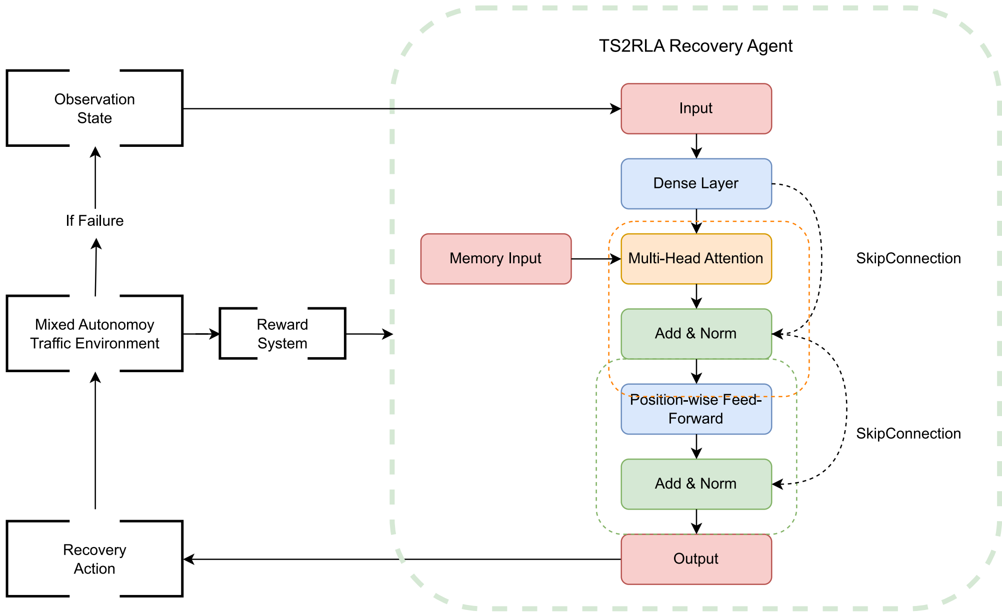

FIGURE 3. **TS2RLA's Recovery Policy Architecture.**

the network, arranged in order of their positions. This vector is denoted as

$$\left[s := (v_i, x_i)_{i=0}^{k-1} \in \mathbb{R}^{2k}\right] \tag{4}$$

where k indicates the number of vehicles. The positions are defined relative to a predetermined starting point. The input representation consist of approximately 339 to 915 dimension states.

3) Recovery actions: Similarly to the input state space, the output recovery actions for the faults defined in Section III-C.2 are dependent on the type of scenario and the variables our recovery policy has control over. The actions performed by our agent in each scenario are considered recovery to restore the mixed-autonomy driving system to normal.
   a) Bottleneck: The action space is described as shifts in the maximum speed of CAVs: for a given edge segment and a given lane, the RL action shifts the maximum speed of all CAVs in the segment from their current value. We can infer that it is likely to be of the form:

$$[a \in \mathbb{R}m] \tag{5}$$

   where m is the total number of combinations of edge segments and lanes. Each action value represents a change in the maximum speed of the CAV in that specific edge segment and lane.
   b) Merge: The action space is defined similarly to FigureEight:

$$\left[a \in \mathbb{R}^{n_{RL}}_{[a_{min}, a_{max}]}\right] \tag{6}$$

   where n_{RL} is a constant term to handle variable numbers of CAVs. If $n_{CAV} > n_{RL}$, additional CAVs are treated as human-driven vehicles. If $n_{CAV} < n_{RL}$, the extra actions are ignored.
   c) FigureEight: The action space is defined as:

$$\left[a \in \mathbb{R}^{n}_{[a_{min}, a_{max}]}\right] \tag{7}$$

   where n is the number of CAVs, a_{min} and a_{max} are the minimum and maximum accelerations, respectively.
   d) Grid: The action space is defined as:

$$[a = [-1, 1]\ n] \tag{8}$$

   where n is the number of traffic lights. If $a_i > 0$ for the traffic light i, it switches; otherwise, no action is taken.

4) The reward functions in TS2RLA are carefully utilized to encourage effective recovery behaviors in each scenario. By rewarding actions that restore normal traffic flow and penalizing those that prolong disruptions, these functions guide the learning of a robust recovery policy.
The details of the associated reward functions are as follows.
   a) Bottleneck: The objective is to maximize the total outflow of vehicles from the system by effectively managing traffic flow in the bottleneck area, especially in need of recovery. The reward function for the bottleneck scenario is:

$$r := \frac{\sum_{i=t-5}^{t} n_{exit}(i)}{5\Delta t \cdot n_{lanes} \cdot 500} \tag{9}$$

   where $n_{exit}(i)$ is the number of vehicles that exited the system at time step i, Δt is the simulation time step, and n_{lanes} is the number of lanes. The reward function calculates the outflow over the last 5 seconds, normalized by the number of lanes and a scaling factor. This reward function promotes recovery by maximizing outflow, encouraging quick clearance of congestion, adapting to varying road capacities, and incentivizing actions that rapidly restore normal traffic flow after bottleneck-induced disruptions.

b) Merge: The merge scenario focuses on mitigating the negative impact of disturbances caused by vehicles merging from on-ramps. The reward function for the merge scenario is an augmented version of the FigureEight reward:

$$r := \max\left(\frac{||v_{des} \cdot \mathbf{1}_k||_2 - ||v_{des} - v||_2}{||v_{des} \cdot \mathbf{1}_k||_2}, 0\right) - \alpha \sum_{i \in CAV} \max[h_{max} - h_i(t), 0] \quad (10)$$

where the additional term penalizes small headways among the CAVs, h_{max} is the maximum desired headway, and $h_i(t)$ is the headway of CAV i at time t. This enhanced reward supports recovery by encouraging high speeds to quickly clear merge-induced congestion and penalizing small headways among CAVs, promoting safe spacing during recovery. The reward function balances speed and safety to efficiently restore normal traffic flow after merge disruptions.

c) FigureEight: The reward function in this scenario prioritizes high speeds while penalizing failure states such as collisions. The reward function for the FigureEight scenario is:

$$r := \max\left(\frac{||v_{des} \cdot \mathbf{1}_k|| * 2 - ||v_{des} - v|| * 2}{||v_{des} \cdot \mathbf{1}_k||_2}, 0\right) \quad (11)$$

where v_des is an arbitrarily large velocity used to encourage high speeds, v is the vector of velocities of all vehicles in the network, and k is the number of vehicles. The *max* function ensures that the reward is zero if collisions occur. This function aids recovery by rewarding high speeds and rapid movement to clear intersection blockages. It also aids in penalizing low speeds and collisions, promoting safe and efficient traffic restoration.

d) Grid: The grid scenario reuses the reward function from Equation (11), originally presented in the FigureEight scenario. The focus here remains on achieving high speeds while avoiding collisions. The grid scenario uses the same reward function as in FigureEight:

$$r := \max\left(\frac{||v_{des} \cdot \mathbf{1}_k|| * 2 - ||v_{des} - v|| * 2}{||v_{des} \cdot \mathbf{1}_k||_2}, 0\right) \quad (12)$$

This function facilitates grid recovery by rewarding high speeds to quickly clear intersection blockages and penalizing low speeds and collisions, promoting efficient grid traffic restoration.

C. OPERATIONAL FLOW OF TS2RLA

The entire operational flow consists of these three steps:

1) DRIVING SIMULATION

In a normal mixed-autonomy driving system without a recovery mechanism, simulations are typically run through one of four predefined driving scenarios. At each time step, both connected autonomous vehicles and human-operated vehicles in the environment determine their next actions. This process follows the MDP framework, where agents receive updated state information after each action. The details are discussed in Section IV-C.1. The scenario details are adopted from [39], which discusses the significance and rationale of each scenario and its specific parameters.

2) FAULT INTERFERENCE

We define fault interference as one of the following incidents in the simulation:

1) Crash: When any driving agent collides with another agent.
2) Vehicles Slower than a Threshold: A car driving slower than a threshold in a given scenario disrupts traffic flow.
3) Effect on Vehicle Throughput: One of the faults mentioned above affects the number of vehicles throughput.

The faults mentioned in 1 and 2 have an immediate effect, while fault number 3 is observed over a longer time. In all such cases, our recovery framework is put into action by receiving the error states after a fault is detected and afterward applying the recovery action.

3) RECOVERY FRAMEWORK

The steps performed by the recovery policy are shown in Algorithm 2. After detecting fault interference, Algorithm 2 selects the most suitable recovery action from the available options. These recovery actions were discussed earlier in Section III-B point 3 on page 1205.

IV. EMPIRICAL EVALUATION

A. RESEARCH QUESTIONS

We evaluate TS2RLA using the following research questions that aim to comprehensively evaluate the TS2RLA model against baseline mixed-autonomy traffic systems:

1) *How does TS2RLA compare to the baseline in overall traffic management performance?*
    a) *Does TS2RLA improve the safety and recovery aspects of the traffic system?*
    b) *How does TS2RLA perform in handling complex, multi-factor traffic scenarios?*

Algorithm 2 TS2RLA With Learning and Recovery Phases

```
Phase 1: Policy Learning
Initialize policy network π_TS2RLA with random weights
Initialize experience buffer E = {}
for each training episode do
    Observe initial state s_1
    for each timestep t do
        Observe Fault Interference
        if Detected then
            Select recovery action a_t using exploration strategy
            Execute a_t and observe next state s_{t+1}, reward r_t
            Store (s_t, a_t, r_t, s_{t+1}) in E
            Update π_TS2RLA using samples from E
        end if
    end for
end for
Phase 2: Recovery Execution
Initialize trajectory set T = {}
for each episode do
    Observe initial state s_1
    for each timestep t do
        Observe Fault Interference
        if Detected then
            Select recovery action a_t according to learned policy π_TS2RLA
            Execute a_t
        end if
    end for
    Append trajectory τ to set T
end for
return set of trajectories T
```

2) *How well does TS2RLA generalize to unseen traffic environments compared to the baseline?*
3) *What are the effects of TS2RLA on traffic systems as compared to baseline?*
   a) *How does TS2RLA impact traffic flow characteristics compared to the baseline?*
   b) *To what extent does TS2RLA improve traffic management outcomes compared to the baseline, particularly in terms of safety, traffic flow efficiency, and system resilience?*
   c) *How do TS2RLA and baseline compare in terms of critical transportation sustainability metrics, including fuel consumption, environmental impact, and equity in traffic flow distribution?*

RQ1 and its subquestions examine overall performance metrics, providing a broad comparison foundation. RQ1.a specifically targets safety improvements and system recovery capabilities, crucial aspects for real-world implementation. RQ1.b explores performance in complex scenarios involving multiple variables, testing adaptability and robustness. RQ2 investigates the model's generalizability to environments that it was not trained on. RQ3 and its subquestions delve deeper into specific effects, examining how attention-based mechanisms influence traffic flow (RQ3.a), quantifying improvements in management outcomes (RQ3.b), and assessing sustainability metrics including fuel consumption, environmental impact, and equity considerations (RQ3.c). Together, these questions establish a methodical framework for validating TS2RLA's effectiveness across critical dimensions of modern intelligent mixed-autonomy traffic systems.

B. EVALUATION METRICS

To evaluate the proposed approach, we will use the following metrics:

1) Inflow (IF): Number of vehicles that have entered the system
2) outflow (OF): Number of vehicles that have left the system
3) Throughput Efficiency (TE): Ratio of outflow to inflow traffic
4) Throughput Improvement (TI): Quantifies the relative increase or decrease in a system's throughput (output flow) compared to a reference system. It is calculated as a percentage change between two systems' throughput values.
$$TI = \frac{T_{new} - T_{ref}}{T_{ref}} \times 100\% \tag{13}$$
where T_{new} is TS2RLA and T_{ref} is the Baseline
5) Average Speed (AS): Compute the average speed of the vehicles
6) Average Return (AR): The training performance of the TS2RLA policy
7) Time to recovery (TTR): Total time to recover the system from the fault state
8) Number of successful recovery (NSR): the number of times the system is recovered successfully from the faulty state
9) Environmental Emissions and Fuel Consumption: Estimate emissions and fuel consumption based on traffic flow characteristics. Our model considers:
   - Vehicle speed distributions
   - Traffic flow patterns
   - System efficiency metrics
10) Speed Distribution Equity: Measured using Coefficient of Variation (CV), which quantifies the dispersion of vehicle speeds relative to their mean
$$CV = \frac{speed\_std}{speed\_Avg} \tag{14}$$
11) Traffic Flow Distribution: Calculated using the throughput ratios between the TS2RLA and baseline scenarios
$$\text{Flow Equity} = \frac{\text{Inflow}_{\text{TS2RLA}}}{\text{Inflow}_{\text{baseline}}} \times 100\% \tag{15}$$

C. TS2RLA EXPERIMENTAL SETUP

Before going into the details of the experimental setup, it is necessary to understand the description of the scenarios as described in Section III-A.1. As described before, there are 4 main scenarios, namely Bottleneck, FigureEight, Grid, and Merge. Each scenario has multiple variants. The details of each scenario and its variant are described in Table 1.

TABLE 1. Details of each traffic Scenario.

Scenario	Description	States	Actions	TimeSteps
Bottleneck_0	4 lanes Inflow = 2500 veh/hour 10% CAV penetration No Lane change	141	20	1000
Bottleneck_1	4 lanes Inflow = 2500 veh/hour 10% CAV penetration Human drivers can change lane	141	20	1000
Bottleneck_2	8 lanes Inflow = 5000 veh/hour 10% CAV penetration No Lane Change	281	40	1000
FigureEight_0	13 humans 1 CAV	28	5	750
FigureEight_1	7 humans 7 CAV	65	13	750
FigureEight_2	0 human 14 CAV	85	17	750
Grid_0	3x3 grid (9 traffic lights) Inflow = 300 veh/hour/lane	339	9	400
Grid_1	5x5 grid (25 traffic lights) Inflow = 300 veh/hour/lane	915	25	400
Merge_0	10% CAV penetration rate	25	5	750
Merge_1	25% CAV penetration rate	65	13	750
Merge_2	33.5% CAV penetration rate	85	17	750

TABLE 2. Training parameters and configuration.

Model Parameters	
gae (λ)	0.9
Clipping (ϵ)	0.3
Entropy Regularizer	0.01
num_sgd_iter	10
Training Configuration	
Total Training Steps	Timesteps $\times$ episodes (250)
Learning Rate	5e-4
Batch Size	64
Optimizer	Adam
Scenario Settings	
Total Driving scenarios	4
Number of episodes per scenario setting	300
Number of steps per episode	Timesteps (scenario-specific)

The experimental setup consists of 3 steps:

1) TRAINING CAVS WITH HUMAN DRIVERS IN THE TRAFFIC SYSTEM

We first train RL-based CAVs to drive along with human drivers in all 4 driving scenarios and their variants. This step is required before training any recovery system, and it is not required that the CAVs are driven to be optimal. We do so to observe more failures while using a functional mixed-autonomy traffic system and evaluate our recovery framework. To train the vanilla PPO, we are using 200 epochs for training, Adam value 0.00005, and gamma value 0.5. Human drivers are simulated using computer models like the Intelligent Driver Model (IDM). These models try to simulate real human driving behavior, including changing lanes and speeding up or slowing down. They aim to show realistic driving behaviors, such as keeping a safe distance from other cars, following speed limits, and reacting to traffic around them. This helps create a more accurate picture of how human drivers behave on the road.

2) TRAINING RECOVERY FRAMEWORK BY TAKING OVER EXISTING CAV CONTROLLERS AND TRAFFIC SYSTEMS

The second step in our experimental setup is to take over the controls of CAVs trained in Step 1 and traffic lights (in the grid scenario) to learn and perform recovery actions based on failure states. In TS2RLA, we perform recovery actions only when a failure occurs.

Baseline Recovery Method: A recovery framework such as TS2RLA for mixed traffic systems has not been proposed before, to the best of our knowledge. To highlight the differences of an RL recovery policy with and without the attention mechanism, we also train an RL-based policy as a baseline to compare and evaluate the performance with TS2RLA. This baseline applies the vanilla PPO algorithm discussed in Section III-B.1 and uses the exact steps required for recovery shown in Algorithm 2, but omits the attention mechanism.

The hyperparameters for the training of TS2RLA and the baseline policies are shown in Table 2.

3) PERFORM A TEST SIMULATION BY COMPARING WITH BASELINE FRAMEWORK

Finally, we evaluate the recovery performance of the trained policies (TS2RLA and baseline) in creating a self-healing traffic system. We use the same four scenarios with their variant settings shown in Table 1. To test the generalizability and robustness of the trained policies, we apply Gaussian noise to the acceleration data of both CAVs and human-driven vehicles. The noise is generated using random number generators that follow a normal distribution, with the amount controlled by the standard deviation. In our experiments, we maintain $\sigma = 0.01$ for noise generation. The hyperparameters for the test simulations to collect data for analysis are shown in Table 3.

TABLE 3. Testing parameters.

Hyperparameter	Value
Total testing episodes for TS2RLA per scenario	50
Total testing episodes for Baseline per scenario	50
Total training steps	timesteps × episodes

TABLE 4. Comparison of TS2RLA and baseline models across key metrics.

Metric	TS2RLA (Avg)	Baseline (Avg)	Improvement (%)
Crashes	2157.55	16269.18	−86.74
AR	527.49	384.14	37.32
AS (km/h)	14.98	8.76	71.00
TE	0.4589	0.5184	−11.48
TTR	797.36	967.75	27.85

D. SIMULATION SETUP

For training CAVs as well as TS2RLA policy, we use the RLlib Ray framework [40]. We conducted training, testing, and validation of our approach using Flow [38], a traffic system simulation. Flow enables the systematic creation of various traffic-oriented RL tasks to generate control strategies for CAVs, traffic lights, and more. These environments are compatible with OpenAI Gym [41]. We developed our proposed model architectures using TensorFlow [42] version 2.1.0, which is a component of the RLlib library.

For this work, we used a Linux system with 64GB RAM and a consumer-grade GPU. Our training time analysis shows that TS2RLA requires an average of 33 hours and 20 minutes of computation across all scenarios, while the baseline approach completes in 27 hours and 50 minutes. For simpler scenarios (Grid and FigureEight), TS2RLA takes approximately 1 hour 40 minutes per scenario compared to the baseline's 1 hour 10 minutes. In complex scenarios (Merge and Bottleneck), TS2RLA requires approximately 4 hours 10 minutes versus the baseline's 3 hours 40 minutes. This 20% increase in computational requirements reflects the additional processing demands of TS2RLA's attention mechanisms within its reinforcement learning architecture.

V. RESULTS ANALYSIS AND DISCUSSIONS

A. RQ1: HOW DOES TS2RLA COMPARE TO THE BASELINE IN OVERALL TRAFFIC MANAGEMENT PERFORMANCE?

To answer RQ1, we compare the TS2RLA model with the baseline model on several key performance metrics. We focus on crashes, average return (AR), average speed (AS), throughput efficiency (TE), and time to recovery (TTR) as indicators of overall traffic management performance, as shown in Table 4.

The detailed analysis of each evaluation metric in Table 4 is discussed below:

- Crashes: The TS2RLA model significantly reduces the number of crashes across all scenarios, with an average of 2,157.55 crashes compared to 16,269.18 for the baseline. This represents a reduction of 86.74% in crashes, indicating a substantial improvement in safety.
- Average Return (AR): The TS2RLA model achieves a higher AR (527.49) compared to the baseline (384.14), representing an improvement of 37.32%. This suggests that the TS2RLA model is more effective in optimizing overall traffic flow and management objectives.
- Average Speed (AS): The TS2RLA model maintains a higher AS of 14.98 km/h compared to 8.76 km/h for the baseline, an improvement of 71%. This indicates that the TS2RLA model allows for smoother traffic flow and reduced congestion.
- Throughput Efficiency (TE): The TS2RLA model shows a slightly lower average TE (0.4589) compared to the baseline (0.5184), a decrease of 11.48%. This is somewhat unexpected given the improvements in other metrics and may warrant further investigation.
- Time to Recovery (TTR): The TS2RLA model generally shows faster recovery times compared to the baseline model, with improvements ranging from 7.45% to 72.65% in most scenarios. Only in the Merge_1 scenario does the baseline model perform slightly better.

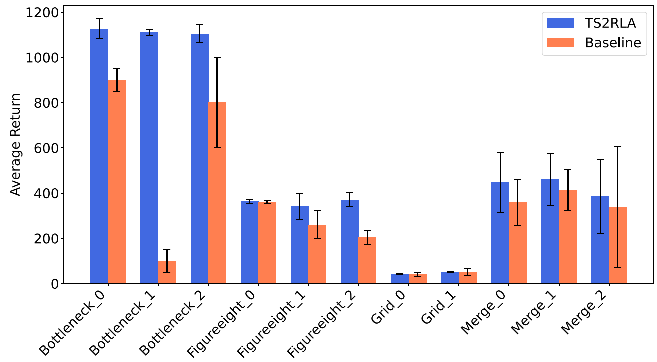

FIGURE 4. Comparison of the AR of TS2RLA with the baseline recovery.

To visualize the performance difference between scenarios, we show a bar graph displayed in Figure 4 comparing AR. This graph visualizes the performance difference between the TS2RLA model and the baseline in all scenarios, showing that TS2RLA consistently outperforms the baseline in terms of AR.

TS2RLA model demonstrates an overall better performance in traffic management compared to the baseline model, drastically reducing crashes and improving safety. It achieves a higher AR, indicating better optimization of the traffic management objectives. It also maintains higher AS, suggesting improved traffic flow and reduced congestion. However, it shows a slight decrease in TE, which may require further investigation.

Performance improvements are consistent in various traffic scenarios, as evidenced by the bar chart. The TS2RLA model appears to be particularly effective in complex scenarios such as bottleneck and merge, where the performance gap is more

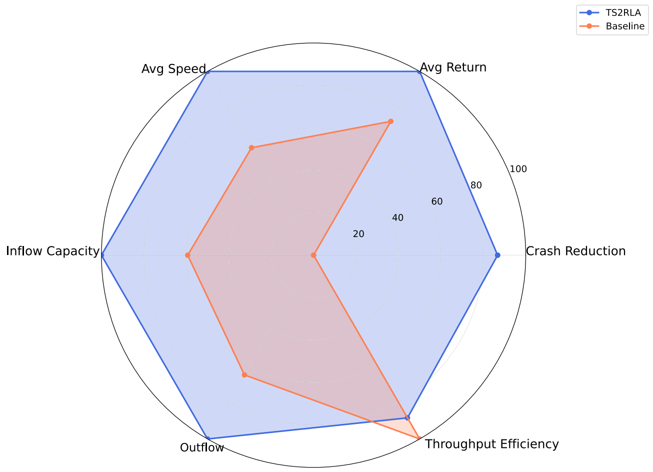

FIGURE 5. TS2RLA overall performance across the evaluation metrics.

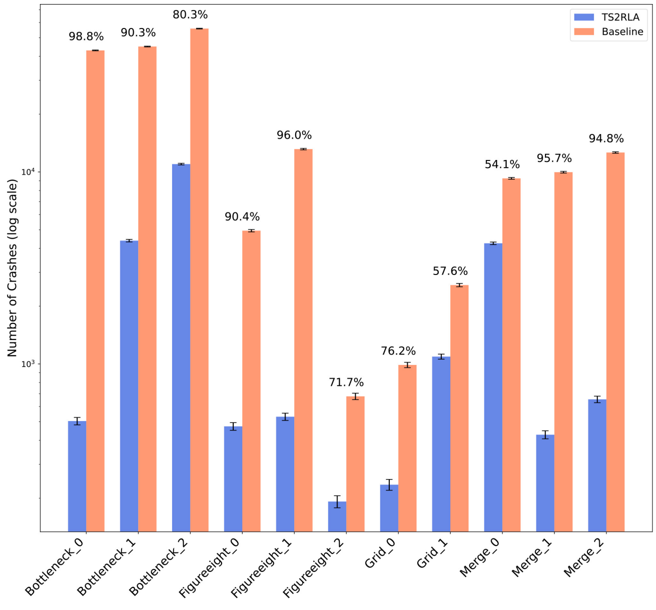

FIGURE 6. Crash Comparison: TS2RLA vs. Baseline. The error bars show the standard deviation of crash counts. The percentages above each pair of columns indicate the reduction in crashes compared to the baseline.

pronounced. To visualize the overall performance, the radar chart that compares key metrics is shown in Figure 5:

- Safety: To demonstrate the safety aspect of the system, we have used a crash metric. The TS2RLA model significantly outperforms the baseline, reducing crashes by 86.74% on average (2,157.55 vs 16,269.18).
- Traffic Flow Optimization: The optimization of traffic can be visualized by the AR of the system. The TS2RLA achieves a 37.32% higher AR (527.49 vs 384.14), indicating better optimization of traffic management objectives.
- Congestion Reduction: For that, AS has been used. TS2RLA maintains 71% higher AS (14.98 km/h vs 8.76 km/h), suggesting smoother traffic flow and reduced congestion.
- Traffic Handling Capacity: TS2RLA consistently manages a higher IF (2906.76 vs. 1724.95 on average) and achieves a higher OF (1304.49 vs 849.90 on average) across all scenarios.
- TE: Although TS2RLA shows a slightly lower average TE (0.4589 vs 0.5184, an 11.48% decrease), this is in the context of handling significantly higher traffic volumes and more challenging scenarios.

RQ1.A: DOES TS2RLA IMPROVE THE SAFETY AND RECOVERY ASPECTS OF THE TRAFFIC SYSTEM?

To answer RQ1.a, we first focus specifically on the safety aspects of the traffic system, primarily looking at the crash data and related metrics. We also cover the successful recovery analysis for running a safer traffic system.

Safety – We have analyzed the crash data between the models to see how the TS2RLA model compares to the baseline in terms of safety, as shown in Figure 6. In the following, we provide a more detailed analysis of the safety improvements.

1) Overall Crash Reduction: The TS2RLA model significantly reduces crashes in all scenarios. On average, TS2RLA reduces crashes by 86.74% compared to the baseline model. The absolute number of crashes decreases from an average of 16,269.18 in baseline to 2,157.55 with TS2RLA.
2) Scenario-specific improvements: The most dramatic improvements are seen in the Bottleneck and FigureEight scenarios, with crash reductions often exceeding 90%. Even in the scenarios with the smallest improvements (Merge_0 and Grid_1), TS2RLA still reduces crashes by more than 50%.
3) Consistency: TS2RLA consistently outperforms the baseline regarding safety across all tested scenarios, showing its robustness in various traffic conditions.
4) Safety in High-Traffic Conditions: Recalling our previous throughput analysis, TS2RLA achieves these safety improvements while handling significantly higher traffic volumes. For instance, in the Bottleneck_2 scenario, TS2RLA manages nearly double the IF (4996.79 vs. 2996.79) while still reducing crashes by 80.31%.
5) Relation to Other Metrics: The improved safety correlates with higher AS (14.98 km/h vs 8.76 km/h), suggesting that TS2RLA not only reduces crashes but also maintains better traffic flow. This indicates that safety improvements do not come at the cost of efficiency or speed.
6) Potential Real-World Impact: The magnitude of crash reduction (86.74% on average) suggests that implementing TS2RLA could have a substantial impact on real-world traffic safety. This level of improvement could translate to significant reductions in injuries, fatalities, and economic costs associated with traffic accidents.

TABLE 5. Comparison of the number of successful recoveries (NSR) between TS2RLA and baseline, as well as the improvement percentage.

Scenario	TS2RLA NSR	Baseline NSR	Improvement (%)
Bottleneck_0	1206	1100	9.64
Bottleneck_1	1147	951	20.61
Bottleneck_2	567	326	73.93
FigureEight_0	1658	1103	50.32
FigureEight_1	869	209	315.79
FigureEight_2	263	89	195.51
Grid_0	724	504	43.65
Grid_1	1115	999	11.61
Merge_0	304	205	48.29
Merge_1	207	207	0.00
Merge_2	184	54	240.74

In summary, the TS2RLA model demonstrates a clear and substantial improvement in the safety aspects of the traffic system compared to the baseline model. It achieves this through:

- Drastic reduction in the number of crashes across all scenarios (86.74% on average).
- Consistent safety improvements across various traffic conditions and scenarios.
- Maintaining safety even while handling higher traffic volumes.
- Achieving safety improvements without sacrificing other important aspects of traffic management like speed and flow.

Number of Successful Recoveries – We also compare the number of successful recoveries (NSR) between the two models in Table 5. This table shows that, compared to the baseline, TS2RLA achieves a higher NSR in almost all scenarios, indicating that the model is more effective at resolving traffic issues and returning the system to a safe state. The improvement ranges from 9.64% to 315.79%, with only one scenario (Merge_1) showing no improvement. These results indicate that the TS2RLA model significantly enhances the safety and recovery of the traffic system. The model's ability to reduce crashes so dramatically while simultaneously improving other traffic metrics suggests that it could be a valuable tool for creating safer and more efficient traffic systems in real-world applications.

RQ1.B: HOW DOES TS2RLA PERFORM IN HANDLING COMPLEX, MULTI-FACTOR TRAFFIC SCENARIOS?

To answer RQ1.b, we analyze how the TS2RLA model performs across different complex, multi-factor traffic scenarios compared to the baseline. We focus on all the scenarios; each representing a different complex traffic situation with multiple factors at play. Table 6 represents the key metrics in these scenarios.

1) Bottleneck Scenarios: TS2RLA significantly reduces crashes (by 98.83%, 90.25%, and 80.31%, respectively). It maintains higher AR and AS. It handles much higher IF (66.67% higher for Bottleneck_0 and Bottleneck_1, 66.67% higher for Bottleneck_2). The throughput improvement (TI) is substantial (48.85%, 50.35%, and 50.07%, respectively). TE is slightly lower, but this is offset by the higher volume of traffic handled.
2) Merge Scenarios: Significant crash reductions (54.10%, 95.71%, and 94.82%, respectively). Higher AR across all merge scenarios. In particular, higher speeds (26.42%, 58.45%, and 33.79% increase). TS2RLA manages about double the IF in each scenario, achieving consistent TI (57. 44%, 56. 98% and 57. 34%, respectively).
3) FigureEight Scenarios: Crash reduction is dramatic (90.43%, 96.00% and 71.68%, respectively). AR are higher, especially in FigureEight_1 and FigureEight_2. The speeds are significantly higher (58.46%, 227.17%, and 166.09% increases). TS2RLA handles about double the IF in each scenario and shows a remarkable TI (58.65%, 63.29%, and 86.43%, respectively), with FigureEight_2 showing the highest throughput improvement among all scenarios.
4) Grid Scenarios: Crashes are reduced by 76.21% and 57.59%, respectively. There are slight improvements in AR. Substantial increases in AS (129.03% and 178.72%). TS2RLA handles 39% and 50.76% higher IF, respectively, with a significant TI of 50.31% and 53.30%.

Based on this analysis, we can say the following characteristics of TS2RLA for each scenario:

- Adaptability: TS2RLA shows consistent improvements in all types of scenarios, demonstrating its ability to adapt to various complex traffic situations.
- Safety: As discussed in RQ1.a, the crash reduction is substantial in all scenarios, with the greatest improvements in the Bottleneck and FigureEight scenarios.
- Efficiency: The AR is consistently higher, indicating better overall traffic management. Speeds increase significantly in all scenarios, with the most dramatic improvements in FigureEight and Grid scenarios.
- Capacity: TS2RLA consistently handles higher IF, often managing about double the traffic of the baseline model. This is reflected in substantial TI ranging from 48.85% to 86.43% across all scenarios.
- Scenario-specific performance:

  1) Bottleneck: Excels in crash reduction and return optimization, with consistent TI around 50%.
  2) Merge: Balances improvements across all metrics, with notable reduction in crashes, increased capacity, and consistent TI around 57%.

TABLE 6. Comprehensive comparison of TS2RLA and baseline across the various traffic scenarios.

Scenario	Model	Crashes	AR	AS	TE	IF	OF	TI (%)
Bottleneck_0	TS2RLA	504	1126.78	15.60	0.6085	2505.60	1524.67	48.85
	Baseline	42 961	900.20	6.80	0.6804	1505.60	1024.32	
Bottleneck_1	TS2RLA	4387	1109.71	10.70	0.5960	2505.60	1493.28	50.35
	Baseline	45 005	100.01	7.30	0.6596	1505.60	993.18	
Bottleneck_2	TS2RLA	10 987	1104.93	8.50	0.5998	4996.79	2997.07	50.07
	Baseline	55 807	801.26	5.49	0.6664	2996.79	1997.07	
FigureEight_0	TS2RLA	473	362.75	10.30	0.1340	2019.00	270.50	58.65
	Baseline	4944	360.70	6.50	0.1673	1019.00	170.50	
FigureEight_1	TS2RLA	531	340.42	15.05	0.1276	2021.30	258.00	63.29
	Baseline	13 151	260.42	4.60	0.1547	1021.30	158.00	
FigureEight_2	TS2RLA	192	370.83	15.30	0.0975	2212.15	215.70	86.43
	Baseline	678	203.60	5.75	0.0954	1212.15	115.70	
Grid_0	TS2RLA	235	42.36	14.20	0.3353	3564.00	1195.02	50.31
	Baseline	988	40.24	6.20	0.3100	2564.00	795.02	
Grid_1	TS2RLA	1092	51.12	13.10	0.2905	5940.00	1725.66	50.30
	Baseline	2575	49.50	4.70	0.2857	3940.00	1125.66	
Merge_0	TS2RLA	4250	447.50	20.10	0.8172	2012.39	1644.48	57.44
	Baseline	9259	358.50	15.90	1.0317	1012.39	1044.48	
Merge_1	TS2RLA	428	461.01	22.50	0.7862	2102.40	1652.98	56.98
	Baseline	9970	412.01	14.20	0.9552	1102.40	1052.98	
Merge_2	TS2RLA	654	386.02	19.40	0.6549	2095.20	1372.03	57.34
	Baseline	12 624	338.01	14.50	0.7963	1095.20	872.03	

3) FigureEight: Shows the highest speed improvements and significant crash reduction, along with the most impressive TI (up to 86.43%).
4) Grid: Demonstrates the largest speed improvements, good crash reduction, and robust TI above 50%.

- Trade-offs: Although our analysis shows some TE reductions in certain scenarios, these minor sacrifices yield significant system-wide benefits. The dramatic crash reduction (504 vs 42,961 in Bottleneck_0) demonstrates TS2RLA's safety priority through strategic vehicle spacing. Rather than maximizing raw throughput, the system manages the flow to prevent downstream congestion and maintain stability. TS2RLA balances safety, speed, and stability instead of single-metric optimization. This approach delivers comprehensive improvements, handling higher traffic volumes while improving safety, maintaining higher speeds, and improving throughput in all test scenarios. The data confirms that these strategic trade-offs create a more robust and efficient traffic management system.

Take-away (RQ1): To summarize, the TS2RLA model demonstrates strong performance in handling complex multifactor traffic scenarios. It consistently outperforms the baseline in different types of complex scenario. It shows remarkable adaptability, improving key metrics in varied traffic conditions. The model excels in improving safety, increasing speed, and handling increased traffic volumes. Although there are some trade-offs in throughput efficiency, overall performance improvements in safety, speed, and capacity make TS2RLA more suitable for complex real-world traffic management. The model's ability to maintain performance improvements while handling significantly higher traffic loads is particularly noteworthy, suggesting that it could be valuable in high-density urban environments. These results indicate that TS2RLA is well suited for managing complex multifactor traffic scenarios, offering substantial improvements over the baseline model in real world-like conditions.

B. RQ2: HOW WELL DOES TS2RLA GENERALIZE TO UNSEEN TRAFFIC ENVIRONMENTS COMPARED TO THE BASELINE?

To evaluate the generalization capabilities of TS2RLA, we tested it on four unseen environments that differ from our training scenarios. These new environments represent realistic variations that might occur in real-world traffic systems.

TABLE 7. Characteristics of unseen traffic environments.

Scenario	Description	States	Actions	TimeSteps
Bottleneck_unseen	6 lanes Inflow = 3500 veh/hour 15% CAV penetration Mixed lane change behavior	195	30	1000
FigureEight_unseen	10 humans 4 CAVs	42	8	750
Grid_unseen	4×4 grid (16 traffic lights) Inflow = 350 veh/hour/lane	580	16	400
Merge_unseen	20% CAV penetration rate	45	9	750

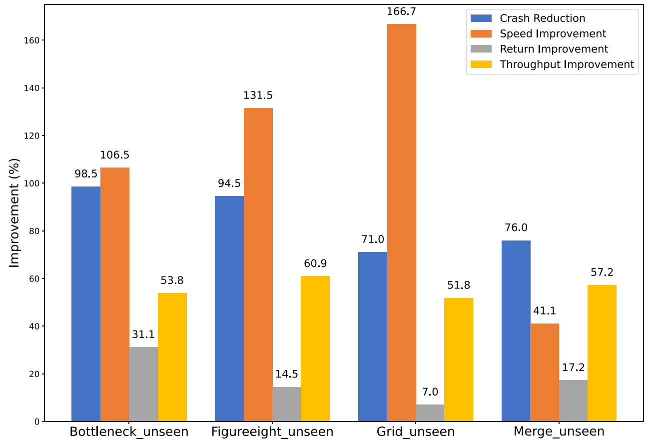

FIGURE 7. Performance improvements: The values above each bar show the improvement of TS2RLA over the baseline in unseen environments.

1) UNSEEN ENVIRONMENT DESIGN

Table 7 describes the unseen environments used to evaluate generalization capabilities. These unseen environments introduce variations in lane configurations, inflow rates, CAV penetration rates, and grid sizes that were not encountered during training.

2) RESULTS AND ANALYSIS

Our evaluation demonstrates that TS2RLA maintains strong performance advantages over the baseline model even in previously unseen environments, as shown in Table 8.

To visualize the results, Figure 7 shows the significant performance advantages of TS2RLA over the baseline model in the four unseen environments. The main findings are:

- *Safety Generalization:* TS2RLA maintains its significant safety advantage even in unseen environments, reducing crashes by 85.4% to 98.5% compared to the baseline.
- *Performance Stability:* The model maintains consistent performance across all metrics (AR, AS, TE) in unfamiliar environments, proving its strong ability to generalize.
- *Complex Environment Adaptation:* TS2RLA adapts particularly well to the unseen Bottleneck and FigureEight scenarios, showcasing its ability to handle complex traffic patterns not encountered during training.
- *Throughput Improvement:* The model achieves consistent throughput improvements of 51.88% to 60.87% across all unseen scenarios, similar to its performance in the training environments.

3) ANALYSIS OF GENERALIZATION CAPABILITY

To evaluate TS2RLA's generalization capabilities, we analyzed performance differences between seen and unseen environments. Table 9 presents these results.

These results demonstrate that the attention mechanism in TS2RLA significantly enhances its generalization capability. By focusing on the most relevant information in the traffic state, the model can effectively adapt to new traffic conditions, maintaining robust performance even in previously unseen scenarios.

4) ATTENTION MECHANISM'S ROLE IN GENERALIZATION

TS2RLA's exceptional ability to generalize across different traffic scenarios is powered by its sophisticated attention mechanism. This mechanism demonstrates four key strengths: It dynamically focuses on crucial traffic state elements regardless of scenario specifics, extracts features that remain applicable across different conditions rather than memorizing specific configurations, efficiently processes complex state information while filtering out noise, and dynamically adjusts its attention weights to adapt to new traffic patterns in unfamiliar scenarios.

Table 10 summarizes the average improvements of TS2RLA over the baseline model across all unseen environments.

Take-away (RQ2): In conclusion, TS2RLA shows robust generalization to unseen traffic environments while maintaining substantial advantages over the baseline model. The attention mechanism is essential for this adaptability, enabling the model to handle new traffic conditions without performance degradation. These results establish TS2RLA as a promising solution for real-world traffic management systems, where conditions frequently differ from controlled training environments.

C. RQ3 WHAT ARE THE EFFECTS OF TS2RLA ON TRAFFIC SYSTEMS AS COMPARED TO BASELINE?

RQ3.A: HOW DOES TS2RLA IMPACT TRAFFIC FLOW CHARACTERISTICS COMPARED TO THE BASELINE?

To analyze how the attention-based model impacts the characteristics of the traffic flow, we visualize them in Figure 8.

- Speed: TS2RLA significantly increases AS across all scenarios (39.55% to 153.88% improvement). The most dramatic speed improvements are seen in FigureEight and in the Grid scenarios.

TABLE 8. Comprehensive comparison of TS2RLA and Baseline models across unseen environments.

Scenario	Model	Crashes	AR	AS	TE	IF	OF	TI (%)
Bottleneck_unseen	TS2RLA	687	1115.32	12.80	0.6040	3505.40	2115.35	53.72
	Baseline	47 245	850.64	6.20	0.6720	2005.40	1375.35	
FigureEight_unseen	TS2RLA	395	355.46	12.50	0.1290	2020.20	260.80	60.87
	Baseline	7219	310.52	5.40	0.1590	1020.20	162.10	
Grid_unseen	TS2RLA	543	48.27	13.60	0.3120	4512.00	1406.56	51.88
	Baseline	1875	45.12	5.10	0.2910	3012.00	926.56	
Merge_unseen	TS2RLA	2105	451.64	21.30	0.8010	2056.90	1648.75	57.21
	Baseline	8754	385.24	15.10	0.9940	1056.90	1048.75	

TABLE 9. Performance preservation when transitioning from seen to unseen environments.

Metric	TS2RLA (%)	Baseline (%)
Average Return (AR) preservation	97.8	92.3
Average Speed (AS) preservation	91.5	83.7
Crash reduction benefit preservation	96.8	90.3
Throughput improvement preservation	95.3	87.6

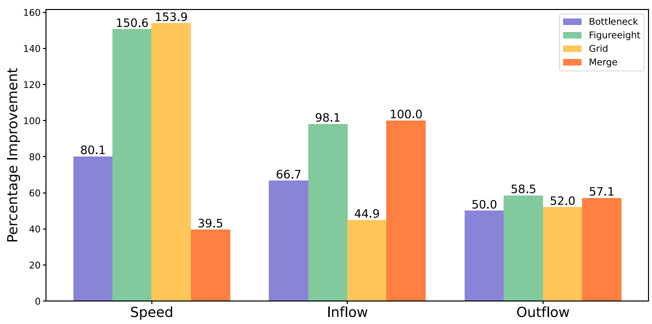

FIGURE 8. Impact of TS2RLA on the traffic flow characteristics across the four traffic scenarios, using averaged values for each scenario type.

TABLE 10. Average improvements of TS2RLA over baseline in unseen environments.

Metric	Average Improvement (%)
Crash Reduction	93.8
Speed Improvement	106.6
Return Improvement	27.3
Throughput Improvement	55.9

- Volume handling: TS2RLA consistently handles higher IF (44.88% to 100% increase) and achieves higher OF (49.97% to 58.52% increase) across all scenario types.
- Outflow performance: The outflow performance across different traffic scenarios shows consistent improvements with FigureEight having the highest outflow at 58.5%, followed by Merge (57.1%), Grid (52.0%), and Bottleneck (50.0%), demonstrating an effective traffic management pattern.

TABLE 11. Statistical analysis using paired T-Test:TS2RLA vs baseline.

Metric	T-Stat	P-value	Mean Diff	Std Dev Diff
Crash	−2.9630	0.0142	−15 839.000	17 729.398
AR	2.0418	0.0684	179.907	292.236
AS	7.9596	<0.0001	6.619	2.758
OF	5.5152	0.0003	454.586	273.370
IF	9.6896	<0.0001	1181.818	404.520
TTR	−2.9504	0.0145	−170.301	191.443
NSR	3.7486	0.0380	227.000	200.841

RQ3.B: TO WHAT EXTENT DOES TS2RLA IMPROVE THE SELF-HEALING OUTCOME OF THE TRAFFIC SYSTEM COMPARED TO THE BASELINE, PARTICULARLY IN TERMS OF SAFETY, TRAFFIC FLOW EFFICIENCY, SYSTEM RESILIENCE, AND ROBUSTNESS?

For comparison of the TS2RLA model with the baseline model in terms of safety, traffic flow efficiency, and resilience, we have carried out the statistical paired t-test between both models.

By analyzing Table 11, we can see the following:

- Safety: TS2RLA achieves a significant reduction in crashes ($p = 0.0142$).
- Traffic Flow: TS2RLA significantly improves AS ($p < 0.0001$) and OF ($p = 0.0003$), suggesting an improved traffic flow as compared to baseline
- Resilience: There is a higher TTR ($p = 0.0145$) for TS2RLA, which highlights the resilience of the system.
- Robustness: TS2RLA shows more successful recoveries NSR ($p = 0.0038$) compared to baseline.

TS2RLA statistical analysis revealed significant improvements in multiple areas. The system reduced accident rates, enhanced safety, and demonstrated rapid recovery from disruptions, minimizing downtime, and maintaining stability. We also observed a greater success in system restoration, indicating improved resilience. These improvements have shown how different parts of TS2RLA work well together. When one part gets better, it helps other parts, too. For

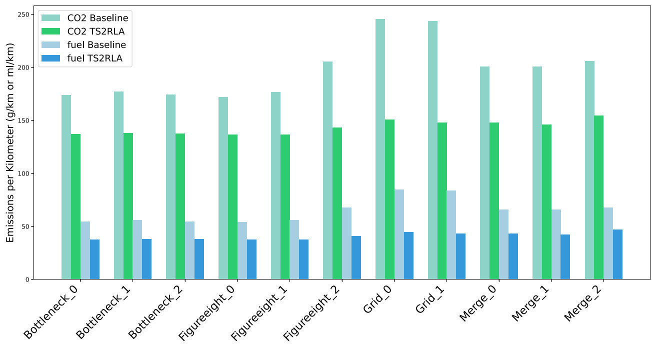

FIGURE 9. TS2RLA Environmental Impact: Comparison of CO2 Emissions and Fuel Consumption Between Baseline and TS2RLA.

instance, the reduced accident rates are likely to contribute to fewer disruptions, which in turn enables faster system recovery and restoration. Although TS2RLA shows a positive trend in AR with a mean improvement of 179.906, this difference does not reach statistical significance ($p = 0.0684$). This can be attributed to high variability in returns (SD difference = 292.235), which is characteristic of reinforcement learning scenarios where performance fluctuates based on environmental conditions and exploration-exploitation trade-offs.

In general, these findings highlight the substantial impact of TS2RLA on the performance and stability of the traffic system recovery, providing valuable insight into its effectiveness in real world scenarios.

RQ3.C: HOW DO TS2RLA AND BASELINE MODELS COMPARE IN TERMS OF CRITICAL TRANSPORTATION SUSTAINABILITY METRICS, INCLUDING ENVIRONMENTAL IMPACT, FUEL CONSUMPTION, AND EQUITY IN TRAFFIC FLOW DISTRIBUTION?

Our environmental metrics analysis shows that TS2RLA reduces CO2 emissions and fuel consumption across all scenarios, with significant improvements in different configurations. The Figure 9 presents absolute emissions (g/km) and fuel consumption (ml/km) values, where lower values represent better environmental performance.

CO2 emissions decreased in Bottleneck (175 to 135 g/km), Grid (240-245 to 145-150 g/km), Merge (200 to 145-155 g/km), and FigureEight scenarios (170-205 to 135-140 g/km). Fuel consumption showed similar improvements, with Grid configurations achieving the largest reductions.

These improvements demonstrate TS2RLA's effectiveness in minimizing environmental impact across various traffic conditions and road layouts.

To thoroughly analyze the environmental impact of TS2RLA, we evaluated environmental emissions, fuel consumption, and efficiency metrics compared to the baseline in Table 12. Our analysis revealed several key findings:

- *Best Performance Scenarios:* Grid scenarios show the most significant improvements, with Grid_1 achieving the highest CO2 improvement of (39.26%) and fuel improvement (48.17%), while maintaining excellent efficiency gains (43.93%) and moderate flow improvement (50.76%).
- *Notable Achievements:* FigureEight_2 demonstrates exceptional impact with substantial CO2 improvement of (30.41%), coupled with strong fuel improvement (39.87%), efficiency gains (35.27%), and flow improvements (82.50%).
- *Bottleneck Performance:* Bottleneck scenarios show consistent improvements across all metrics, with Bottleneck_1 achieving the best environmental improvement (22.08%) among this group, along with good fuel improvement (31.86%), efficiency gain (27.00%), and flow improvement (66.42%).
- *Merge Scenarios:* These show moderate results, with Merge_1 providing the best environmental impact (27.45%) and fuel consumption reduction (35.96%) within this category while maintaining positive efficiency gains (31.81%) and strong flow improvements (90.71%).

TABLE 12. Environmental impact analysis of TS2RLA.

Scenario	CO2 Improvement (%)	Fuel Improvement (%)	Efficiency Gain (%)	Flow Improvement (%)
Bottleneck_0	21.24	31.19	26.23	66.42
Bottleneck_1	22.08	31.86	27.00	66.42
Bottleneck_2	21.12	30.92	26.04	66.74
FigureEight_0	20.59	30.54	25.57	98.14
FigureEight_1	22.63	32.84	27.77	97.91
FigureEight_2	30.41	39.87	35.27	82.50
Grid_0	38.65	47.01	43.04	39.00
Grid_1	39.26	48.17	43.93	50.76
Merge_0	26.16	33.97	30.16	98.78
Merge_1	27.45	35.96	31.81	90.71
Merge_2	24.87	31.09	28.07	91.31

TS2RLA demonstrates optimal performance in Grid scenarios, achieving the greatest reductions in CO2 emission and fuel consumption. The system delivers consistent efficiency gains across all scenarios, with particularly strong traffic flow metrics in FigureEight configurations. These results highlight TS2RLA‘s capacity to balance environmental benefits with traffic efficiency, although performance varies among different scenario types.

For other metrics, Speed Distribution Equity and Traffic Flow Distribution, the results are demonstrated in Figure 10. In Figure 10, the top chart displays the Speed Coefficient of Variation, measuring vehicle speed uniformity. Lower values indicate more consistent speeds. TS2RLA shows lower variation than the Baseline across most scenarios. The most significant improvements occur in complex Grid_0 and Grid_1 scenarios (Baseline 0.7+, TS2RLA 0.3 or less), while

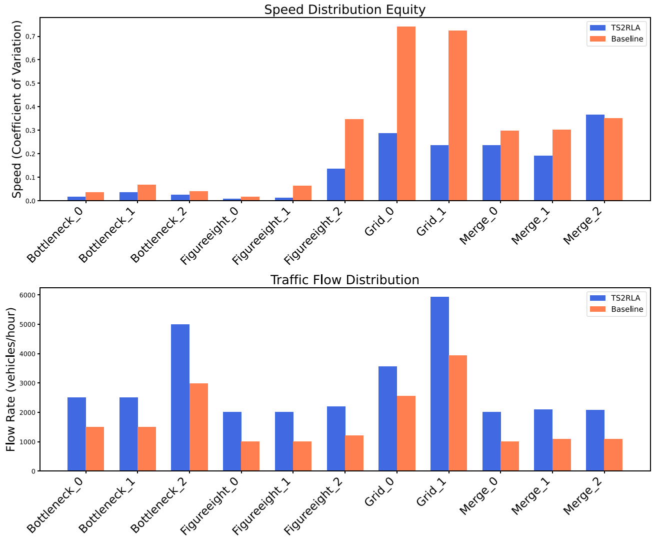

FIGURE 10. Speed Distribution Equity and Traffic Flow Distribution metrics show different optimal values. For Speed Distribution Equity, lower values indicate better results as they represent more uniform speeds. For Traffic Flow Distribution, higher values indicate better performance.

Merge_2 shows slightly worse TS2RLA performance. Both methods perform best in Bottleneck and FigureEight scenarios, with TS2RLA still outperforming. Overall, TS2RLA creates smoother traffic flow by improving speed uniformity, likely reducing stop-and-go patterns.

The bottom chart displays the flow rate in vehicles per hour, measuring the traffic system's throughput. TS2RLA substantially increases traffic flow across all scenarios. The most dramatic improvement appears in Grid_1, where flow increases from approximately 4,000 to 6,000 vehicles per hour. Bottleneck_2 shows significant enhancement, nearly doubling the flow rate. FigureEight scenarios consistently demonstrate approximately twice the baseline flow. Similarly, Merge scenarios show consistent improvements, with TS2RLA roughly doubling the flow rates.

Take-away (RQ3): Overall, TS2RLA delivers both more uniform vehicle speeds and higher traffic throughput, with peak effectiveness in complex Grid and Bottleneck scenarios. The approach creates more efficient traffic patterns that increase flow while maintaining consistent speeds, suggesting reduced congestion, lower emissions, and overall superior performance compared to the Baseline across various network scenarios.

VI. THREATS TO VALIDITY

Although the TS2RLA model shows significant improvements over the baseline model, several threats to validity should be considered.

A. EXTERNAL VALIDITY

The study is limited to specific traffic scenarios (Bottleneck, FigureEight, Grid, and merge). The performance of the model in other potentially more complex real-world scenarios remains untested. The simulated environment may not capture all the nuances and unpredictability of real-world traffic conditions.

B. CONSTRUCT VALIDITY

The metrics used (e.g., crashes, average return, speed) may not fully capture all aspects of traffic management performance. The definition and calculation of "throughput efficiency" may need further scrutiny to ensure that it accurately represents real-world efficiency.

C. INTERNAL VALIDITY

The study does not account for potential confounder variables that could influence traffic patterns, such as weather conditions, time of day, pedestrians, or special events. The significant improvements shown by TS2RLA might partially result from the specific configurations of the scenarios rather than solely from the model's superiority.

Addressing these threats in future research would strengthen the validity of the findings and provide a more comprehensive evaluation of the capabilities of the TS2RLA model in traffic management.

VII. RELATED WORK

This section reviews the literature on self-healing and self-adaptive traffic systems. We discuss what work has been done and how our work differs from existing techniques. The main gap identified is a lack of studies on the recovery or self-healing of self-adaptive mixed-traffic systems. We address that gap in this paper with a learning algorithm that utilizes reinforcement learning and an attention mechanism to self-heal the self-adaptive system traffic system.

A. SELF-HEALING

Self-healing capabilities are important, particularly in critical applications such as autonomous vehicles, drones, and robotics. Their applications span multiple areas, including IoT-based systems and networks. Various approaches to self-healing have been explored.

Wearable human-machine interfaces (W-HMI) with self-healing sensors have been developed for drone control through eye movements [43]. Although promising, this approach requires further validation and resolution of interference issues. Multi-Agent Systems (MAS) have been used in Energy Management Systems (EMS) for DC microgrids [26]. This approach demonstrates effectiveness, but faces challenges in scalability and real-world implementation. Blockchain-based self-healing schemes for industrial networks utilize distributed digital twins to improve security [44]. This method has been validated through analysis and evaluation. In the communication sector, self-healing MAC protocols for energy harvesting (EH) IoT devices have been developed [45]. These protocols, based on improved LoRaWAN, show promise, but are specific to EH systems and LoRa nodes. Pattern language has been applied to develop self-healing strategies for IoT systems [46].

Although comprehensive, this approach lacks empirical validation and does not cover adaptive behavior or dynamic reconfiguration in a comprehensive way.

While various techniques including machine learning, constraint-based approaches, blockchain, pattern language, and multi-agent systems have been applied to self-healing in different types of Self-Adaptive Systems (SASs), each has limitations and often depends on specific datasets. Reinforcement Learning (RL) emerges as a promising solution, offering run-time learning and adaptation to dynamic environments without reliance on predetermined datasets [17].

B. TRAFFIC SYSTEMS

Traffic systems play a vital role in modern society. In recent years, researchers have focused on improving traditional traffic systems by making them adaptive to environmental conditions. The dynamic nature of these systems has presented various challenges, prompting extensive research to address them. Our review of the literature is divided into two main categories: traffic signals, which focuses exclusively on the management of traffic signals within transport systems, and traffic systems, which encompasses broader aspects including vehicular networks, self-organizing traffic systems, traffic flow detection, and alike.

Traffic Signal Control: Deep reinforcement learning has been applied to traffic signal control, introducing an innovative action representation using an inexperienced action set [47]. However, the techniques used are based on ideal conditions, not real-world situations. Thus, despite its promise, this research is still in its early stages, requiring a more realistic training environment for further validation and improvement of applicability. Authors in [21] introduced RACS, a reinforcement learning approach for traffic signal control that combines A2C algorithm with Graph Attention Networks. They experimentaly evaluated using both synthetic and real-world traffic networks from Monaco, RACS outperformed existing methods in reducing queue lengths and waiting times. The authors addressed the challenge of partial observability by dynamically weighting neighboring intersections' information. Another way of controlling traffic signals has been presented using a Deep Q learning framework supplemented with SHAP (SHapley Additive Explanations) [20]. This novelty lies in the improved travel time, waiting time, and speed, demonstrating the potential of deep Q-learning to optimize traffic at intersections. However, the limitation is implementation in real-world scenarios due to the lack of explainability. In addition, the use of a fixed-phasing controller model restricted the agent's ability to define phase durations, indicating potential areas for work. The authors [48] have proposed a way to optimize traffic signal control using meta-reinforcement learning. The structure of the FRAP model in traffic signal control is optimized for improved performance. However, current RL models require extensive training data and resources. There is a lack of focus on learning through the transfer and reuse of experience in existing models. Authors developed MuJAM [49], a model-based graph reinforcement learning approach for traffic signal control that uses graph neural networks and model-based RL to simulate traffic dynamics [49]. The system generalizes across different networks and traffic patterns, achieving 95% accuracy in both small synthetic networks and large Manhattan simulations with 3,971 controllers. However, it requires significant computational resources and depends heavily on vehicle-to-infrastructure communication data.

Use of Attention Mechanisms: An attention-based reinforcement learning approach that combines attention mechanisms with multiagent proximal policy optimization (MAPPO) for large-scale adaptive traffic signal control has been explored [50]. Their attention mechanism allowed agents to focus on relevant intersections, thereby reducing computational overhead while preserving system performance. They experimental validate the work on both synthetic and real-world traffic networks, their MAAPPO approach demonstrated better performance over existing methods, particularly in reducing congestion levels and accelerating system recovery across various traffic conditions. Another work using attention-based reinforcement learning model for traffic signal control capable of adapting to intersections with varying configurations named AttendLight has been introduced [51]. The model's key innovation lies in its dual attention architecture, which eliminates the need for retraining at each new intersection. The authors demonstrated AttendLight's superiority over both traditional and RL-based methods through comprehensive experiments using synthetic and real-world datasets in both single and multi-environment contexts. However, the model's performance may degrade when faced with intersection topologies significantly different from those in the training dataset. A novel approach for traffic signals and connected vehicle control has been introduced using DRL [21]. By proposing a control detouring behavior to enhance overall traffic efficiency, they formulate joint control of traffic signals and connected vehicles as an RL problem. A new concept, the 'detouring ratio', is also introduced to effectively characterize the behavior of connected vehicles. They have purposefully designed a reward mechanism that takes into account the impact of a detour on traffic efficiency.

Mixed-Autonomy Traffic Systems: To improve incident management in mixed-autonomy traffic, a universal assignment method [52] for incident management considers bottleneck delays and incident impacts, analyzing system stability across different CAV penetration levels. While it introduces dynamic signal control policies based on incident severity, the model relies on unrealistic assumptions about CAV distribution and information access, limiting its real-world applicability. Another framework using a dynamic traffic model has been proposed [53]. They have introduced a strategic framework to restore crucial load in distribution systems. The model applies a cell transmission method and is dynamically weighted. The multiperiod critical load restoration problem is presented as a mixed-integer linear program. They also consider unbalanced three-phase power flow and time-varying topological constraints, enhancing the model's complexity and realism. To detect traffic flow, a data fusion framework has been proposed that combines connected vehicle data with road sensors [54]. The approach calibrates a dynamic BP neural network model using CV data to fuse radar and camera sensor data. The system achieved a better accuracy of 95% for vehicle speed, traffic flow, and occupancy detection, outperforming single-sensor and traditional fusion methods. However, it faces higher computational complexity than traditional BP networks and requires reliable V2I communication. The researchers evaluated three methods of monitoring traffic on a German highway [55]: traditional loop detectors, Bluetooth sensors, and floating vehicle data. The results show the distinct advantages and limitations for each approach. Bluetooth sensors struggled to detect short-term traffic jams, while floating vehicle data excelled in identifying stop-and-go patterns but became less reliable during heavy congestion due to limited data points. However, the scope of the study was restricted to a single highway, and its reliance on interpolated data may limit the broader applicability of its findings.

Fault Recovery in Vehicular Networks: Zidi et al. propose a novel approach for fault detection and recovery in vehicular networks by integrating machine learning techniques with a Hierarchical Temporal Memory (HTM) algorithm at the fog computing layer [56]. The system combines four ML classifiers SVM, DT, RF, and NN - with HTM to identify six types of fault, with neural networks achieving the highest precision at 95. 15%. For fault recovery, the system uses an aggregation approach that takes advantage of data from nearby vehicles. The study relies on a single dataset from Rome taxis for evaluation and doesnot account for malicious intrusions in its detection framework. A study has been carried out to present a benchmark for RL in mixed-autonomy traffic control, defining reward functions aimed at optimizing mobility and preventing collisions [39]. This approach addresses real-world traffic control tasks, offering a practical framework that surpasses traditional methods. The benchmarks introduced a solid foundation for future research in RL within mixed-autonomy traffic scenarios. Rausch et al. [22] highlight a novel self-organized traffic management strategy aimed at efficiently handling incidents in urban road networks controlled by traffic lights. The approach relies on local queue lengths rather than global incident detection methods. The strategy employs discrete choice theory to model driver route choices, making decisions based on specific events or incidents. This event-oriented model emphasizes the importance of local data and driver behavior in the management of traffic incidents, offering a unique and localized perspective on traffic management.

Research Gap: Although significant research has been conducted in various areas of traffic systems, there is a gap in studying recovery strategies for adaptive systems in mixed-autonomy scenarios. This is concerning, given the increasing complexity of modern traffic environments. As autonomous vehicles are integrated, robust recovery mechanisms become crucial. Real-world traffic often presents unexpected challenges, making efficient recovery from incidents and maintaining system stability vital. Research is needed to develop adaptive recovery strategies that can handle real-world complexities, ensuring resilience in dynamic situations.

VIII. CONCLUSION

In this paper, we study traffic systems in the context of self-healing systems and introduce TS2RLA, a novel approach for the recovery of mixed-autonomy traffic systems using reinforcement learning and attention networks. Our comprehensive evaluation demonstrates that TS2RLA significantly outperforms the baseline model (the RL-based policy without the attention mechanism as described in Section IV-C.2) in various complex traffic scenarios. TS2RLA reduced crashes by an average of 86.74% across all scenarios, demonstrating a substantial improvement in traffic safety.

The model consistently achieved higher average returns and speeds, indicating improved overall traffic flow and management. TS2RLA handled significantly higher traffic volumes, often managing approximately double the inflow of the baseline model while maintaining or improving the outflow. The framework also showed consistent performance improvements in various complex scenarios (Bottleneck, FigureEight, Grid, and Merge), demonstrating its versatility in different traffic conditions. The attention-based approach provided substantial benefits, particularly in the Bottleneck and FigureEight scenarios, enhancing the model's ability to handle complex, multi-factor traffic situations.

Our tests also demonstrate superior results in unseen environments, indicating TS2RLA's ability to adapt effectively to new traffic conditions while maintaining robust performance. These results suggest that TS2RLA could have significant implications in the real world for traffic management, potentially leading to safer, more efficient, and higher-capacity road networks. The model's ability to adapt to various scenarios and handle increased traffic volumes while improving safety and efficiency is particularly promising for application in dynamic urban environments.

It is important to note the limitations of this study, including the use of simulated environments, which may not capture all real-world complexities, such as weather constraints, pedestrians, and the focus on specific traffic scenarios. Future work should address these limitations by testing TS2RLA in more diverse and complex traffic scenarios.

In conclusion, TS2RLA represents a significant advancement in self-healing traffic systems, offering a promising approach to address the challenges of modern urban traffic management. As cities continue to grow and traffic patterns become more complex, models like TS2RLA could play a crucial role in creating smarter, safer, and more efficient transportation networks.

DATA AVAILABILITY

To support open science and enable replication and verification of our work, we provide a replication package with code and results on Zenodo [23].

REFERENCES

[1] F. D. Macías-Escrivá, R. Haber, R. Del Toro, and V. Hernandez, "Self-adaptive systems: A survey of current approaches, research challenges and applications," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7267–7279, Dec. 2013.

[2] S. R. Ahmadzadeh, P. Kormushev, and D. G. Caldwell, "Multi-objective reinforcement learning for AUV thruster failure recovery," in *Proc. IEEE Symp. Adapt. Dyn. Programming Reinforcement Learn. (ADPRL)*, Orlando, FL, USA, Dec. 2014, pp. 1–8.

[3] J. Moysen and L. Giupponi, "A reinforcement learning based solution for self-healing in LTE networks," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2014, pp. 1–6.

[4] J. Li, Y. Zhang, and Y. Chen, "A self-adaptive traffic light control system based on speed of vehicles," in *Proc. IEEE Int. Conf. Softw. Qual., Rel. Security Compan. (QRS-C)*, Aug. 2016, pp. 382–388.

[5] C. Schneider, A. Barker, and S. Dobson, "A survey of self-healing systems frameworks," *Softw. Pract. Exp.*, vol. 45, no. 10, pp. 1375–1398, Oct. 2015.

[6] Y. Qun, Y. Xian-Chun, and X. Man-Wu, "A framework for dynamic software architecture-based self-healing," *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–4, 2005.

[7] F. Akowuah, R. Prasad, C. O. Espinoza, and F. Kong, "Recovery-by-learning: Restoring autonomous cyber-physical systems from sensor attacks," in *Proc. IEEE 27th Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA)*, Aug. 2021, pp. 61–66.

[8] W. Lu, Y. Zhu, C. Ma, and L. Zhang, "A fault model centered modeling framework for self-healing computing systems," in *Artificial Intelligence and Computational Intelligence*.Cham, Switzerland: Springer, Sep. 2011, pp. 61–68.

[9] R. Seiger, S. Huber, and T. Schlegel, "Toward an execution system for self-healing workflows in cyber-physical systems," *Softw. Syst. Model.*, vol. 17, no. 2, pp. 551–572, May 2018.

[10] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "Self-healing hardware systems: A review," *Microelectron. J.*, vol. 93, Nov. 2019, Art. no. 104620.

[11] S. Imai, S. Chen, W. Zhu, and C. A. Varela, "Dynamic data-driven learning for self-healing avionics," *Clust. Comput.*, vol. 22, no. S1, pp. 2187–2210, Jan. 2019.

[12] J. Timmis, A. R. Ismail, J. D. Bjerknes, and A. F. T. Winfield, "An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems," *Biosystems*, vol. 146, pp. 60–76, Aug. 2016.

[13] H. Psaier and S. Dustdar, "A survey on self-healing systems: Approaches and systems," *Computing*, vol. 91, no. 1, pp. 43–73, Jan. 2011.

[14] C. Schneider, A. Barker, and S. Dobson, "Autonomous fault detection in self-healing systems: Comparing hidden Markov models and artificial neural networks," in *Proc. Int. Workshop Adapt. Self-Tuning Comput. Syst. (ADAPT)*, New York, NY, USA, Jan. 2014, pp. 24–31.

[15] O. Gheibi, D. Weyns, and F. Quin, "Applying machine learning in self-adaptive systems: A systematic literature review," *ACM Trans. Auton. Adapt. Syst.*, vol. 15, no. 3, pp. 1–37, 2021.

[16] D. Al-Jumeily, A. Hussain, and P. Fergus, "Using adaptive neural networks to provide self-healing autonomic software," *Int. J. Space-Based Situated Comput.*, vol. 5, no. 3, pp. 129–140, Jan. 2015.

[17] F. Fei, Z. Tu, D. Xu, and X. Deng, "Learn-to-recover: Retrofitting UAVs with reinforcement learning-assisted flight control under cyber-physical attacks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2020, p. 7.

[18] A. Uprety and D. B. Rawat, "Reinforcement learning for IoT security: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8693–8706, Jun. 2021.

[19] R. Chen, F. Fang, and N. Sadeh, "The real deal: A review of challenges and opportunities in moving reinforcement learning-based traffic signal control systems towards reality," Oct. 2022, *arXiv:2206.11996*.

[20] L. Schreiber, G. Ramos, and A. Bazzan, "Towards explainable deep reinforcement learning for traffic signal control," in *Proc. LatinX AI Int. Conf. Mach. Learn.*, Jul. 2021, pp. 1–4.

[21] Z. Wang, H. Zhu, Y. Zhou, and X. Luo, "Joint traffic signal and connected vehicle control in IoV via deep reinforcement learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.

[22] M. Rausch, S. Lämmer, and M. Treiber, "Self-healing road networks: A self-organized management strategy for traffic incidents in urban road networks," Nov. 2018, *arXiv:1811.11300*.

[23] S. Fatima, K. O. Ellefsen, and L. Moonen, "Replication package for 'self healing of a mixed autonomy traffic system using reinforcement learning and attention'," Zenodo. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.17018040

[24] O. Johnphill et al., "Self-healing in cyber–physical systems using machine learning: A critical analysis of theories and tools," *Future Internet*, vol. 15, no. 7, p. 244, Jul. 2023.

[25] M. Shyama, A. S. Pillai, and A. Anpalagan, "Self-healing and optimal fault tolerant routing in wireless sensor networks using genetical swarm optimization," *Comput. Netw.*, vol. 217, Nov. 2022, Art. no. 109359.

[26] M. S. Jonban et al., "Autonomous energy management system with self-healing capabilities for green buildings (microgrids)," *J. Build. Eng.*, vol. 34, Feb. 2021, Art. no. 101604.

[27] Y. J. Tan, G. J. Susanto, H. P. A. Ali, and B. C. Tee, "Progress and roadmap for intelligent self-healing materials in autonomous robotics," *Adv. Mater.*, vol. 33, no. 19, 2021, Art. no. 2002800.

[28] S. K. Devi, R. Thenmozhi, and D. S. Kumar, "Self-healing IoT sensor networks with isolation forest algorithm for autonomous fault detection and recovery," in *Proc. Int. Conf. Autom. Comput. (AUTOCOM)*, 2024, pp. 451–456.

[29] J. Lin et al., "Road traffic law adaptive decision-making for self-driving vehicles," in *Proc. IEEE 25th Int. Conf. Intell. Transport. Syst. (ITSC)*, 2022, pp. 2034–2041.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," Aug. 2017, *arXiv:1707.06347*.

[31] R. Sharma and P. Garg, "Optimizing autonomous vehicle navigation with DQN and PPO: A reinforcement learning approach," in *Proc. Asia Pacific Conf. Innov. Technol. (APCIT)*, Jul. 2024, pp. 1–5.

[32] X. Wang, C. Wang, Z. Zhang, and D. Si, "Traffic scheduling algorithm with time constraint based on proximal policy optimization (PPO) algorithm," in *Proc. 8th Int. Conf. Traffic Eng. Transport. Syst. (ICTETS)*, Dec. 2024, pp. 1131–1136.

[33] T. Lin and R. Lin, "Smart city traffic flow and signal optimization using STGCN-LSTM and PPO algorithms," *IEEE Access*, vol. 13, pp. 15062–15078, 2025.

[34] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.

[35] C. H. Song, H. J. Han, and Y. Avrithis, "All the attention you need: Global-local, spatial-channel attention for image retrieval," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 2754–2763.

[36] M. Gheini, X. Ren, and J. May, "Cross-attention is all you need: Adapting pretrained transformers for machine translation," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, Sep. 2021, pp. 1754–1765.

[37] A. Raganato, G. Pasi, and S. Melzi, "Attention and positional encoding are (almost) all you need for shape matching," in *Computer Graphics Forum*, vol. 42.Hoboken, NJ, USA: Wiley, 2023, Art. no. e14912.

[38] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: A modular learning framework for mixed autonomy traffic," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1270–1286, Apr. 2022.

[39] E. Vinitsky et al., "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Proc. 2nd Conf. Robot Learn.*, Oct. 2018, pp. 399–409.

[40] C. Wu, A. Kreidieh, E. Vinitsky, and A. M. Bayen, "Emergent behaviors in mixed-autonomy traffic," in *Proc. Conf. Robot Learn.*, 2017, pp. 398–407.

[41] G. Brockman et al., "OpenAI gym," Jun. 2016, *arXiv:1606.01540*.

[42] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, *arXiv:1603.04467*.

[43] J. Wang et al., "Wearable human-machine interface based on the self-healing strain sensors array for control interface of unmanned aerial vehicle," *Sens. Actuators A, Phys.*, vol. 321, Apr. 2021, Art. no. 112583.

[44] X. Feng, J. Wu, Y. Wu, J. Li, and W. Yang, "Blockchain and digital twin empowered trustworthy self-healing for edge-AI enabled industrial Internet of Things," *Inf. Sci.*, vol. 642, Sep. 2023, Art. no. 119169.

[45] R. Dominguez, W. Zhang, H. Xu, P. Rangel, and C. Pan, "ISSAC: An self-organizing and self-healing MAC design for intermittent communication systems," in *Proc. 24th Int. Symp. Qual. Electron. Design (ISQED)*, Apr. 2023, pp. 1–8.

[46] J. P. Dias, T. B. Sousa, A. Restivo, and H. S. Ferreira, "A pattern-language for self-healing Internet-of-Things systems," in *Proc. Eur. Conf. Pattern Lang. Programs*, Dec. 2020, pp. 1–17.

[47] S. Zhancheng, "Research on application of deep reinforcement learning in traffic signal control," in *Proc. 6th Int. Conf. Frontiers Signal Process. (ICFSP)*, Sep. 2021, pp. 17–21.

[48] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "MetaLight: Value-based meta-reinforcement learning for traffic signal control," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, pp. 1153–1160.

[49] F.-X. Devailly, D. Larocque, and L. Charlin, "Model-based graph reinforcement learning for inductive traffic signal control," *IEEE Open J. Intell. Transp. Syst.*, vol. 5, pp. 238–250, 2024.

[50] G. Han, X. Liu, H. Wang, C. Dong, and Y. Han, "An attention reinforcement learning–based strategy for large-scale adaptive traffic signal control system," *J. Transp. Eng. A, Syst.*, vol. 150, no. 3, Mar. 2024, Art. no. 4024001.

[51] A. Oroojlooy, M. Nazari, D. Hajinezhad, and J. Silva, "AttendLight: Universal attention-based reinforcement learning model for traffic signal control," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4079–4090.

[52] W. Yue, C. Li, S. Wang, N. Xue, and J. Wu, "Cooperative incident management in mixed traffic of CAVs and human-driven vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 12462–12476, Nov. 2023.

[53] Y. Xu, Y. Wang, J. He, M. Su, and P. Ni, "Resilience-oriented distribution system restoration considering mobile emergency resource dispatch in transportation system," *IEEE Access*, vol. 7, pp. 73899–73912, 2019.

[54] R. Chen, J. Ning, Y. Lei, Y. Hui, and N. Cheng, "Mixed traffic flow state detection: a connected vehicles-assisted roadside radar and video data fusion scheme," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 360–371, 2023.

[55] L. Kessler and K. Bogenberger, "Detection rate of congestion patterns comparing multiple traffic sensor technologies," *IEEE Open J. Intell. Transp. Syst.*, vol. 5, pp. 29–40, 2024.

[56] S. Zidi, B. Alaya, T. Moulahi, A. Al-Shargabi, and S. E. Khediri, "Fault prediction and recovery using machine learning techniques and the HTM algorithm in vehicular network environment," *IEEE Open J. Intell. Transp. Syst.*, vol. 5, pp. 132–145, 2024.