IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.XXXX/XXXXX.2021.DOI

Deep Reinforcement Learning based Driving Strategy for Avoidance of Chain Collisions and its Safety Efficiency Analysis in Autonomous Vehicles

ABU JAFAR MD MUZAHID[1], SYAFIQ FAUZI BIN KAMARULZAMAN[2] (SENIOR MEMBER, IEEE), MD. ARAFATUR RAHMAN[3] (SENIOR MEMBER, IEEE), ALI H ALENEZI[4]

[1]Faculty of Computing, Universiti Malaysia Pahang,Malaysia, 26600, (e-mail: mrumi98@gmail.com)
[2]Faculty of Computing, Universiti Malaysia Pahang,Malaysia, 26600, (e-mail: syafiq29@ump.edu.my)
[3]School of Mathematics and Computer Science, University of Wolverhampton, UK (e-mail: arafatur.rahman@wlv.ac.uk)
[4]Remote Sensing Unit, Northern border university, Arar, Saudi Arabia, (e-mail: ali.hamdan@nbu.edu.sa)

Corresponding author: Syafiq Fauzi Bin Kamarulzaman (e-mail: syafiq29@ump.edu.my).

The authors would like to thanks to Ministry of Higher Education of Malaysia for promoting this researh and Universiti Malaysia Pahang for providing the labratory facility and financial support under Fundamental Research Grant Scheme(FRGS/1/2018/TK08/UMP/02/2)

ABSTRACT Vehicle control in autonomous traffic flow is often handled using the best decision-making reinforcement learning methods. However, unexpected critical situations make the collisions more severe and, consequently, the chain collisions. In this work, we first review the leading causes of chain collisions and their subsequent chain events, which might provide an indication of how to prevent and mitigate the crash severity of chain collisions. Then, we consider the problem of chain collision avoidance as a Markov Decision Process problem in order to propose a reinforcement learning-based decision-making strategy and analyse the safety efficiency of existing methods in driving security. To address this, A reward function is being developed to deal with the challenge of multiple vehicle collision avoidance. A perception network structure based on formation and on actor-critic methodologies is employed to enhance the decision-making process. Finally, in the safety efficiency analysis phase, we investigated the safety efficiency performance of the agent vehicle in both single-agent and multi-agent autonomous driving environments. Three state-of-the-art contemporary actor-critic algorithms are used to create an extensive simulation in Unity3D. Moreover, to demonstrate the accuracy of the safety efficiency analysis, multiple training runs of the neural networks in respect of training performance, speed of training, success rate, and stability of rewards with a trade-off between exploitation and exploration during training are presented. Two aspects (single-agent and multi-agent) have assessed the efficiency of algorithms. Every aspect has been analyzed regarding the traffic flows: (1) the controlling efficiency of unexpected traffic situations by the sudden slowdown, (2) abrupt lane change, and (3) smoothly reaching the destination. All the findings of the analysis are intended to shed insight on the benefits of a greater, more reliable autonomous traffic set-up for academics and policymakers, and also to pave the way for the actual carry-out of a driver-less traffic world.

INDEX TERMS Autonomous Vehicles, Deep Reinforcement Learning Method, Reward Function, Chain Collision Avoidance, Autonomous Traffic Flow, Safety Efficiency Analysis.

I. INTRODUCTION

AUtonomous vehicles (AVs) are currently regarded as the technology for revolutionizing the existing modes of travel. The research shows that AVs have the potential to improve road safety by eliminating human errors [1] and by optimising traffic congestion [2].The security and reliability of AVs is regarded as one of the most important criteria that must be met before mass production and deployment in real-world traffic systems [3].The analysis of the collision risk of AVs and of autonomous traffic flow is an essential phase in the development of AVs, and it has attracted considerable attention in recent years [4]. For example, the European NCAP

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access

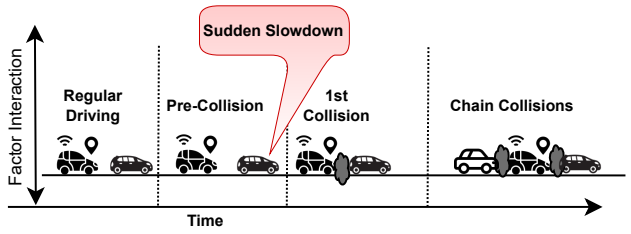


FIGURE 1. Illustration of four consequence phases of chain collision in AVs.

(European New Car Assessment Program) has developed a number of test scenarios to assess the adaptability of AV control logic in the context of collision avoidance [5].

Another key research area is collision prevention among multiple vehicles. The majority of earlier research focused on two-dimensional route-planning issues in the setting of a group of AVs attempting to prevent stationary objects [6]. Recently, experts have highlighted the necessity of automobile collision avoidance for multi-agent systems. In some attempts [7], [8], for each vehicle agent, other road participants vehicles are treated as movable obstacles. By projecting observed velocities, one can estimate where other cars will be in the next state and prevent collisions accordingly. This solitary approach, however, is incapable of imposing limitations on the continuous speed of vehicles or their turn radius [9]. As a result, what appears to be a safe manoeuvre in the present time step may result in crashes in the coming state. Vehicles may be forced to change course in an instant due to the kinematic limits of the vehicles involved. However, this is not practicable in many actual circumstances [10]. In other studies, parametric curves are used to simulate the path to ensure that all movable objects in the environment can get smooth diversions and eventually reach their projected destination [11], [12]. However, vehicles must constantly adjust their speed and positioning to trace these routes [13], and the consequence of the shift is rather large, which is not feasible. It presents a potential direction for overall issues in the safety efficiency analysis of the collision avoidance scheme of AVs.

Although numerous advances have been made in the field of leading vehicle–follower vehicle traffic flow control, the challenge of chain collision avoidance has yet to be thoroughly explored. In dynamic environment, chain collision avoidance and mitigation of its severity among multiple agents, becomes increasingly difficult. There are three main types of conventional algorithms for avoiding collisions: offline planning, artificial potential field approaches, and the sense-and-avoid technique. However, the computational cost of these approaches is high. It is also inconvenient to implement in a dynamic traffic environment because the entire environment's information must be known ahead of time. The current state of the art in these methods can be divided into two categories: reaction-based methods and prediction-based techniques. Due to the fact that they do not take into account future states, all reaction-based approaches are limited in their scope and may be unreliable in some critical situations [14]. However, there are two issues that must be addressed, however: the estimating inaccuracy caused by a variety of uncertainties and the high computational complexity required to perform the prediction operation as a result of the different uncertainties. To address the limitations imposed by traditional collision avoidance processes, numerous researchers have been working on techniques centered on deep learning as well as deep reinforcement learning (DRL). In recent years, deep neural networks (DNNs) have demonstrated their ability to extract features, classify objects, and grasp complicated scenes. However, this strategy only takes into account static obstacles. To depict a complicated traffic scenario using DNN, one needs high-dimensional features as well as a huge amount of training data (samples). Moreover, huge amounts of data, especially in emergency traffic situations, are incredibly difficult to gather. Due to these constraints, recent research has focused on time-dependent back propagation as Recurrent Neural Network (RNN) based models for aggregated data through time to help the agent learn to adopt an optimal driving policy based on both the present and prior observations. There are several algorithms in RRN, such as basic RNN, Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU). We selected LSTM for this experiment for the following reasons: 1) In comparison to other RRNs, LSTM has less computational cost and is better at extracting temporal features while also being able to deal with the difficulty of training without a substantial amount of samples [15]. 2) The LSTM structure supports direct sensor inputs such as camera and 2D-lidar data.This includes the high-accuracy map, traffic data, and the planner-centric objective location stated in polar coordinates. 3) DRL uses LSTM to discover an optimal driving policy, while the feature extraction layer uses supervised learning to train the network in the feature extraction layer. And finally, LSTM has an advantage over hidden Markov decision models as well as the sequence learning approaches since it is less sensitive to the length of the gaps between the data points.

This study seeks to provide a chain collision avoidance strategy and its safety efficiency analysis process for autonomous traffic flow [16] equipped with deep reinforcement learning in uncertain traffic flows, notably in the two severe scenarios of quick deceleration and lane shift, [12], to resolve collision avoidance challenges [17]. In particular, we strive to achieve an advanced method for taking decisions by analyzing the weaknesses and limitations of driving manoeuvres supported by existing DRL algorithms that consider a range of specific action approaches, including lane change, lane retention, velocity maintenance, acceleration, and braking. *Sugiyama et al.* [18] used the optimum velocity model criterion for the collision to study factors which influence the chain collisions in traffic flow. *Figure:1* depicts the impact with the consequences of a sudden slowing scenario on the chain collision. *Z. Li et al.* [19] investigated the likelihood of a chain collision induced by an abrupt stop in view of the

impact of the modern intelligent control system. The affect of the velocity friction-dependent on the multiple car collision was explored by *Nagatani et al.* [20]. In our prior work [21], we presented a learning-based potential risk assessment method to forecast multiple vehicle collision risks as a means of solving this critical problem.

The developed systematic safety efficiency measurement [22], [23] process has two primary aspects, namely the modelling of uncertainty [24] and the safety efficiency analysis. In the phase of uncertainty modeling, we proposed an approach to avoiding chain collisions that is considered as an MDP (Markov Decision Process) that might be solved by applying DRL [25]. In the phase of safety efficiency analysis, we conducted an in-depth investigation of existing RL algorithm-based [26] decision-making methods parameter impacts [27]. During the training phase, we investigate the active learning function to efficiently perform the collision avoidance scheme safety efficiency analysis. The summary of these aspects allows us to evaluate the accurate collision avoidance scheme safety of AVs control rationality in a precise and efficient way. The possible applicability of the suggested method in the safety efficiency analysis based design optimization of controlled parameters is reviewed in accordance with the developed collision-avoidance efficiency analysis.

Therefore, the research contributions are summarized as:

- This article proposes a driving strategy to address the issue of multiple-vehicle collisions. Two critical scenarios are viewed as a new, comprehensive challenge as opposed to the design of a technique for avoiding this type of collision. The agent will learn how to balance driving behavior by interacting with the environment and making the right decision in these two critical situations.
- A rewarding and punishment structure is being developed to address the challenge of multiple vehicle collision avoidance. A perception network structure based on formation, as well as actor-critic methodologies, is also used to enhance the decision-making process.
- To demonstrate an accurate safety efficiency analysis, we used Unity3D to deploy the driving strategy in both single-agent and multi-agent simulation environments, and we used three cutting-edge Deep Reinforcement Learning algorithms to compare which one is optimal.

The remaining sections in this article are: *Section II* is the literature review that will provide a detailed perception of chain collision avoidance in AVs. *Section III* reveals the basic insights of DRL algorithms focused on chain collision-avoidance strategy and the safety efficiency analysis aspects. *Section IV* represents the simulation implementation procedure of the proposed approach, results and safety efficiency analysis and finally *Section V* renders the concluding remarks.

II. RELATED WORK

The majority of current research goes considerably beyond the single autonomous vehicles (AVs) control aspect. In reality, the demands of diverse driving scenarios and the emergence of robust embedded systems, sensors, and networks have led to a broader interest in the subject of cooperative motion control by multiple vehicles. The challenges of making decisions on vehicle control are generally split into three sections for a single autonomous vehicle: 1) localization and surrounding mapping, 2) trajectory tracking, and 3) path planning. And for multiple vehicles, forming a coordinated path generation for multiple vehicles is a major challenge. Each vehicle, in particular, takes a collision-free path, and all vehicles arrive at their individual destinations. Environment sensing results are crucial for intelligent vehicle decision-making and control because they influence how the vehicle operates [28]. When it comes to modeling the vehicle surrounding environment, a robust perception framework is only the responsible. The situation of the ego-vehicle (i.e., speed, position, , and heading) as well as data collected by sensors are fed into the system for this function [29]. Now a days, Cooperative perception is now being developed further, with the aim of sharing not just one's own state information but also the objects captured by other participants. With regard to future technological developments, the ultimate target is to make a means of combining data from on-board sensors with data from other external sources, allowing for the creation of a high-level understanding of the surrounding that encompasses both navigable space and objects. In [30], *N.Sugiyama et al.* analyze and depict a region map for single, double, triple, and chain vehicle collision scenarios in the context of unexpected deceleration. The *authors* [31] in the situations of automatic and manual driving, which are restricted for safety reasons, to evaluate the stringent steadiness of multiple vehicles. In fact, a series of unsteady connected vehicles is more likely to cause a chain collision.

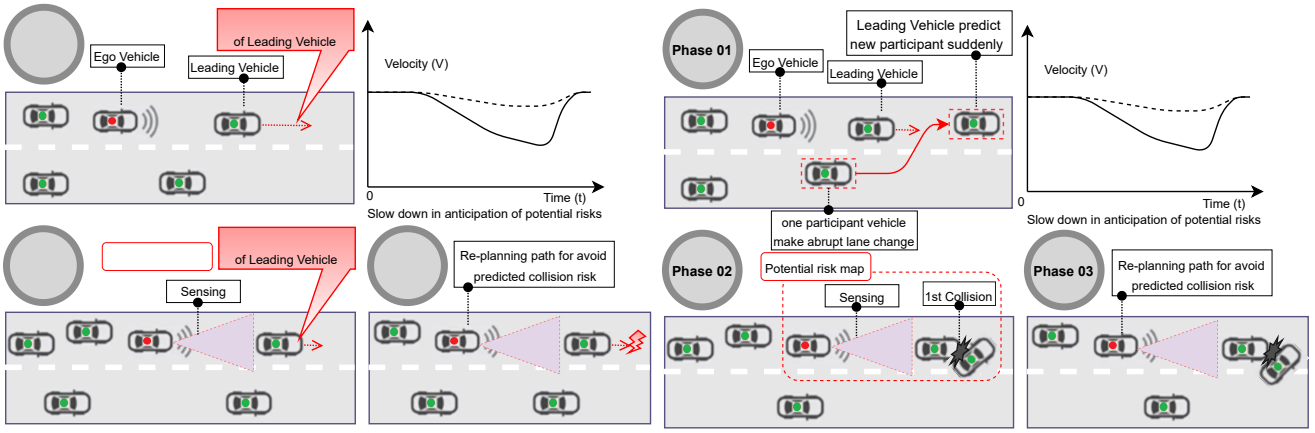
1) Chain Collision

Drivers on the road sometimes rely largely upon the tail brake lights of the leading vehicle to decide whether they need to slow down by braking or not. Conversely, multi-vehicle collisions occur when a car shifts from its own lane to the next lane on a two-lane roadway. These generate potentially dangerous scenarios when a vehicle closely follows another, especially when it is only possible to look behind the vehicle in front of the vehicle. The driver's reaction time is usually 0.85 to 1.6 seconds between the onset and frequency of the braking [32]. If narrow vehicle-to-vehicle distances are maintained to prevent collisions in abrupt braking situations, there may be little margin of defense [33]. In addition, in heavy traffic flows, the cumulative reaction times of subsequent drivers would result in a number of secondary incidents and multi-vehicle collision chains [34].

2) Traffic situation ontology

The optimum understanding of environmental sensing it mostly results crucial [29] of the ontology of highway traffic mostly depends on the perception of uncertain traffic conditions in a group of vehicles, and it is investigated in [35]. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access



(a) Potential risk prediction of chain collision by sudden blockage.

(b) Potential risk prediction of chain collision by abrupt lane change.

FIGURE 2. The depiction of traffic situation ontology by representing the risk mapping and path re-planing scheme of ego vehicle in the aspect of chain collision avoidance in AVs.

circumstances in which vehicles are involved must be understood. This is the basis for other various implementations, including sophisticated chain collision detection and chain collision mitigation schemes. The benefit of understanding the surroundings would enable multiple vehicles to operate autonomously at high speeds in dangerous circumstances and in complex highway or inner-city surroundings or cooperative maneuvers [36]. Typically, the first collisions disrupt traffic flow, block the road, and create severe congestion. The blocking sometimes causes secondary or chain collisions. They are one of the worst traffic collisions, most of which occur on high-speed and high-capacity highways, including expressways. *Figure:*2 depicts the traffic situation ontology of chain collisions and their prevention or mitigation.Where*Figure:*2a and *Figure:*2b represent, respectively, the potential risk assessment and path re-planning to avoid chain collisions caused by sudden blockage and abrupt lane change.

3) Chain Collision Avoidance Techniques

For autonomous vehicles, traffic situations with multiple vehicles interacting are difficult. Even if another traffic participant's rough intent is understood, all participating vehicles must agree on a coordinated and conflict-free motion plan, indirectly or explicitly. For each vehicle, the movement must be secure and comfortable, and it must accommodate all individual goals and desires, [37]. We may calculate the average interval of all these distances by realizing the collision-free distance for each participating agent. When all of the agent velocities are chosen at the same time, conflicts will almost definitely be avoided due to velocities at non-intersecting distances. In a common interval, the selection of competing speeds involves the agreement protocol [38]. Collision detection and avoidance in agents [39] or multi-agent scenarios [40] have also been discussed as a navigation query. Considering the contemporary achievements and identifying the challenges and flows, we concerned ourselves with the

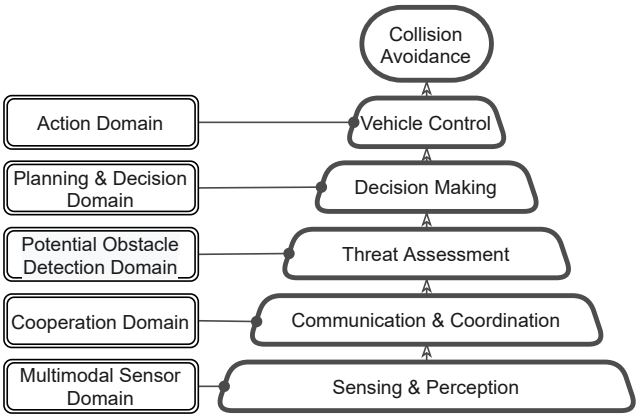


FIGURE 3. Taxonomy of Chain collision avoidance in AVs

remedies to form a taxonomy of Chain Collision Avoidance in AVs, as shown in *figure:*3.

Theoretically, chain accidents can be avoided or decreased in severity by reducing the time between an emergency occurrence and the moment when approaching vehicles are informed of it. Propagating a vehicle-to-vehicle incident warning alert is one way to do this. This could make it possible to circumvent the usual chain of drivers responding to the activation of vehicle brake lights immediately ahead of them and even allow drivers to react to an incident before seeing it. This function is hereinafter referred to as the CCA (Car Accident Avoidance). The secondary crash mitigation strategies are discussed in [41]. Common strategies are employed for chain collision avoidance systems as Platooning, Active Brake Control, Time-Critical Cooperative Control, Trajectory Re-planning. *U. Z. A. Hamid et al.* [42] proposed a new chain crash avoidance technique in *Figure:*2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access

III. METHODOLOGY

In order to address the subsequent chain events of an autonomous vehicle chain collision as well as the traffic situation ontology, we will examine the problem of collision avoidance as a Markov Decision Process that can be addressed using DRL in this section. Our earlier study [43] compared two DRL methods as the foundation for selecting the DRL methodology in this work for more in-depth examination and investigation. In comparison to previous approaches such as mathematics and physics-based methods, chain collision avoidance applications using DRL do not necessitate the use of a significant mathematical model. As an alternative, a competent model was created automatically by altering the parameters of a neural network (NN) that was centred on the observations obtained from the sensors as inputs. Furthermore, this strategy enables the robot to work efficiently in the absence of an accurate map or a high-quality sensor. To evaluate the safety efficiency of the DRL based decision making process immediately, it is indeed an elaborate discussion of the insights of DRL algorithms. The mathematical details of the training environment in simulation and the implementation specifics of DRL-centred collision avoidance approach are described in the subsections that follow.

A. REINFORCEMENT LEARNING (RL) AND MARKOV DECISION PROCESS (MDP)

It is possible to portray RL as a coherent mathematical form for solving the problems of sequential decision-making by interacting with both the agent and the surrounding environment. This is how an agent learns about the environment around it. It adjusts its behaviour in response to the input (a reward or a penalty) it receives as a result of its actions. Considering the foregoing, the real-world environment for RL can be thought of as a Markov Decision Process (MDP).

Typically, an MDP consists of four components, which are represented by $\langle S, A, T, r\rangle$ as a tuple, in here the S and A are state space action space. The transition function which can express as $T : S \times A \rightarrow S$, and the reward function can be identified as $r : S \times A \rightarrow R$. At each stage of the sequential process of this decision-making, the agent takes different actions and alters the state of the environment. The environment then rewards (or punishes) the agent. Afterwards, the procedure is repeated as necessary until the episode comes to an end. When it comes to making decisions, the aim of an agent is to find an optimal set of actions (called a policy) that maximises the expectation of cumulative reward $R = \sum_{t=0}^{T} \gamma^t r^t$, where γ and T denotes as a discount factor, and the experimental time horizon respectively.

In fact, an agent can figure out the optimal policy by performing iterative searches in a small state space. Since state space expands, standard RL methods are limited in their ability to deal with it as the dimensional explosion problem arises. Even if deep learning has made it possible to resolve issues of additional complexity of dimensions and, the DRL approaches have been developed in response. DRL tackles the issue of dimensional explosion by using a deep neural network to approximate the optimal value function. Furthermore, the approach of replay memory and the target network is implemented in order to reduce the correlation of state distribution, resulting in a satisfactory level of performance. As a result, DRL has found widespread use in a variety of fields, including video games, self-driving cars, and even healthcare. Also noteworthy is the fact that DRL is considered a major technique for artificial intelligence.

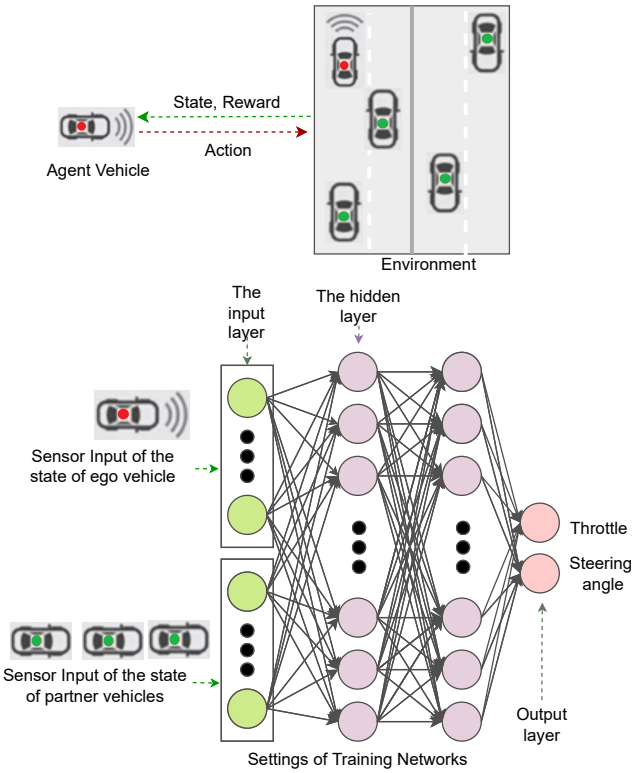


FIGURE 4. Proposed Decision-Making Model

B. PROPOSED MODEL

This work proposes a driving strategy and investigates the safety efficiency of decision-making with the aid of Deep Reinforcement Learning (DRL) in uncertain traffic flows to resolve chain collision avoidance difficulties among multiple autonomous vehicles, especially in the two severe scenarios of rapid deceleration and lane shift. In particular, we strive to achieve an advanced method for taking decisions by analyzing the weaknesses and limitations of driving behaviours supported by existing DRL algorithms that consider a range of specific action approaches, including lane change, lane retention, velocity maintenance, braking, and acceleration. In this study, we define our autonomous vehicle driving decision-making strategy to avoid the chain collision problem corresponding to the MDP. In order to find an optimal action policy that is represented by an actor network, as shown in *Figure 4*, we propose our strategy. There are sensors on both the ego vehicle and all of its participants. The

network, called an actor, takes this information and generates optimal outputs as a result of driving decisions. The optimal outputs are then passed on by the actor network to a controller in the ego vehicle.

$State\ space(S)$: We consider the state space of the environment as, $S = (x_i, y_i, d_{x,i}, d_{y,i};\ i \in 0, ...n_{vehicle})$ A traffic scenario is made up of a set of vehicle positions (x_i, y_i) and respective speeds $(d_{x,i}, d_{y,i})$ where 0 index corresponds the ego vehicle. Within the range $x_b = 20m$, a controlling agent can see the state of all vehicles in surrounding.

$Action\ space(A)$: At each time step, the agent has the option of selecting any combination of lateral and longitudinal actions. These actions are "$stay\ in\ lane$", "$change\ left$," and "$change\ right$" for lateral action, and set acceleration to "1, 0, $-1m/s^2$" for longitudinal movement. The fourth option is to set the longitudinal deceleration $-4m/s^2$ and apply the hard brake. In total, there are a total of 10 different alternative actions. A lane shift cannot be reversed after it has begun. It is configured to "$stay\ in\ lane$" laterally and to travel at $-4m/s^2$ longitudinally as the fallback action.

The aim of the ego vehicle is to develop a policy that maximizes the cumulative reward over time. Usually, we can express the policy π a probability function of the state space $\pi : S \rightarrow (A = a|S)$. Our system, However, the policy as a function that relates states and actions. The accelerator and steering values of the ego vehicle can be determined by the policy based on the present state values of the environment. In our proposed DRL model, a neural network is used to embody the policy.

To avoid collisions, the agent must collect data from the surroundings (considering the above state space $S = \{s_1, s_2, ...s_n\}$, and action space $A = \{a_1, a_2, ...a_{10}\}$) utilising sensors and produce directives to prevent obstacles. A mathematical model can reflect the association between sensor observations and action. Generally, *equation:*1 and *equation:*2 can be used to define the problem.

For single agent approach,

$$d_t, u_t = f b_t \tag{1}$$

where d_t, u_t and b_t represent the linear velocity, the angular velocity, and the sensor observation at each of the time steps, respectively.

For multiple agent approach,

$$d_{t,i}, u_{t,i} = f b_{t,i}\ ; i = 1, 2, 3, ...n \tag{2}$$

where $d_{t,i}$, $u_{t,i}$ and $b_{t,i}$ are the *i-th* number of agents linear velocity, the angular velocity, and the sensor observation at each of the time steps, respectively. By this approach we will evaluate the safety efficiency of DRL in terms of existing three algorithms.

C. DRL METHODS

DRL is an approach that combines reinforcement learning with deep learning. Based on the current policy, an agent can generate the training data by itself in the learning scheme of reinforcement learning through the interaction with its environment. Therefore, there is a continual change in observation and reward data distribution as the agent gains experience, which might produce instability in the learning procedure. Basically, the neural network is fed by the observations of the agent, which outputs the action based on the current policy in the deep reinforcement learning approach. Hyperparameter adjustments are extremely sensitive to these methods. PPO overcomes all these challenges by being simple to tune and implement. It directly learns from the data that it gets from the present states of its surroundings as an on-policy gradient technique, rather than through the Q-learning process used by DQN (deep Q-Networks); the entire learning procedure is based on such data, which is offline data that has been acquired before. When evaluating the performance and trustworthiness of these DRL algorithms, it is important to understand how they work. The algorithm PPO has two parts: policy gradient loss and trust-region. These will be explained first.

In PPO approaches, the *equation:* 3. is policy gradient loss defined initially, which allows for an increase in positive-rewarding actions and a decrease in negative-rewarding ones.

$$\mathcal{L}^{pg}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(\alpha_t|s_t) \hat{B}_t \right] \tag{3}$$

here $\pi_\theta(\alpha_t|s_t)$ denotes the action α_t executed in time t with a policy parameter π_θ given state s_t. The loss function $\mathcal{L}^{pg}(\theta)$ is represented here as the estimated reward of the carried out action at time t, while the second term $\hat{\mathbb{E}}_t$ and a advantage function $\hat{B}_t$ represents the comparative cost function of the prudently carried out action at time t. In the procedure of Critic architecture, use of the Generalized Advantage Estimator (GAE) is very common method to calculate the Advantage function $\hat{B}_t$) as:

$$\hat{B} + (\gamma\varepsilon)\hat{B}_{t+1} + ... + \gamma\varepsilon^{N-t+1}, \tag{4}$$

With

$$\hat{B} = r_t + \gamma V_\theta(S_{t+1}) - V_\theta(S_t) \tag{5}$$

Where, γ and ε are discount factor and GAE estimator parameter respectively and $t = 0, 1, 2, \ldots, N$.

The advantage function is usually divided into two parts: the first one is the discounted sum of rewards, and the second one is the baseline estimate. The discounted sum of rewards in *equation:* 6the weighted sum of all rewards obtained by the agent during each and every timestep in the current episode.

$$\sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{6}$$

Here γ is the discount factor, mostly laid between of 0.9 to 0.99, implies that rewards in close proximity are prioritised over rewards provided further ahead. It calculates all values of rewards limited to t to $t + \infty$. The awards are compounded through a discount factor equivalent to same number of timesteps forward. Because of advantage function is determined once the episode sequence is gathered from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access

the environment, all the rewards are then known. The fact that the reward function is calculated following the episode sequence from the environment is known to all rewards. The second portion of the reward function is the neural network baseline function. The calculation of the baseline function is a discounted return estimate from the position it holds now. Depending on the previous experience, it forecasts the expected reward from the ending of every episode. The neural networks input is each state, and its output is the sum of the discounted anticipated rewards.

A difficulty with the gradient descent is the excessive update of the parameters, creating a policy that is inadequate for the agent to collect suboptimal data. When updating a policy, TRPO makes sure that it is not too out of the way from the prior policy.

$$maximize_\theta \hat{\mathbb{E}}_t \left[\frac{\pi_\theta\left(\alpha_t|s_t\right)}{\pi_{\theta pre}\left(\alpha_t|s_t\right)} \right] \hat{B}_t$$

$$subject\ to\ \hat{\mathbb{E}}_t \left[CL\left[\pi_\theta pre(.|s_t), \pi_\theta(.|s_t)\right]\right] \leq \delta \qquad (7)$$

Because here all the policies are stochastic, the action selected for them is called $\pi_\theta(.|s)$. The CL constraint is added to the goal to prevent over-fitting, and the quantity of policy shift is constrained. The limited CL, on the other hand, limits the optimization strategy and might occasionally result in undesirable training behaviours. Due to the need to diminish the issue in *equation:*7, this new constraint is explicitly mentioned in PPO.

1) PPO

Consider $r_t(\theta)$, which represents the new-to-old policy ratio. The ratio value greater than 1, means the action is more probable now than under the old policy. It is possible to multiply the ratio $r_t(\theta)$ and the objective function for making better readability as *equation:*8.

$$\mathcal{L}^{CPI}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta\left(\alpha_t|s_t\right)}{\pi_{\theta pre}\left(\alpha_t|s_t\right)} \right] \hat{B}_t = \hat{E}_t \left[r_t(\theta)\hat{B}_t \right] \qquad (8)$$

Maximizing the $\mathcal{L}^{CPI}(\theta)$ would cause in overly massive policy changes if there was no constraint. As a result, an objective function could be changed as *equation:*9 to punish policy changes that shift $r_t(\theta)$ from 1.

$$\mathcal{L}^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta)\hat{B}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{B}_t \right] \qquad (9)$$

Here the ϵ denotes hyperparameter, and the value supposed ϵ = 0.2. $\hat{\mathbb{E}}$ represents the expectation operator of the objective function that PPO optimizes calculated by lots of experiences. This operator of expectations shall be accepted for at least two terms. The first one is $r_t(\theta)\hat{\hat{B}}_t$ guarantees the policy adopts actions that give the baseline a high favorable advantage. $clip((r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{\hat{B}}_t$ is the second term, removes the chance for r_t to migrate away from the interval $[1-\epsilon, 1+\epsilon]$ over clipping the likelihood ratio, which affects

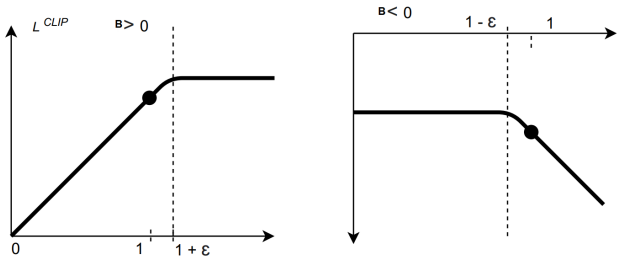


FIGURE 5. The effect of the advantage function on the clipping functionality.

the aim. The positive or negative value estimation may affect the operator's effect.

The positive or negative value of advantage function belongs to the limited probability ratio, as shown in *figure* 5. During each cycle, each N agent, in this example, only 1, acquires T timesteps data. All the experience is gathered. The policy gradient is then done for every lot of K epochs on the policy network to adopt the policy and use a limited PPO target.

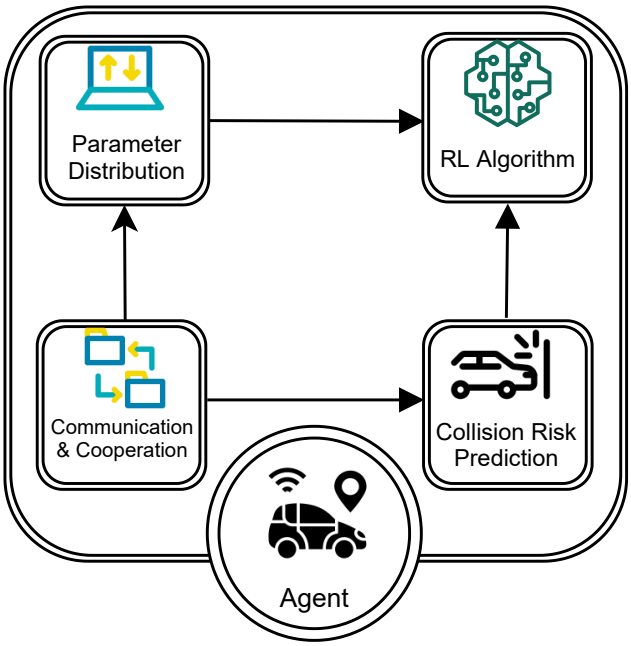


FIGURE 6. *Presentation of the deep reinforcement learning agent.*

2) Soft Actor-Critic

The Soft Actor-Critic (SAC) method is an off-policy technique with entropy regularisation as a core component. Measuring the randomness of policies, SAC policy seeks to maximise the equilibrium between an anticipated return and entropy. The situation is analogous to the trade-off between exploration and exploitation, which might further enhance the learning rate and prevent policy convergence early. To illuminate soft actor-critique, the setting of reinforcement

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access

learning that poses entropy-regularized learning is first introduced. Entropy can be evaluated as the amount of randomness of a variable. The probability function X will be as follows in the case of considering the x as a random variable, and this distribution function P defines F the entropy of random variable x in *equation:*10.

$$H(P) = \mathop{\mathbb{E}}_{x\sim p}[-\log P(X)] \tag{10}$$

In reinforcement learning, entropy-regularized learning, a bonus in line with entropy, is granted to the agent in every timestep, adjusting the problem of reinforcement learning.

$$\pi^* = arg\max_{\pi} \mathop{\mathbb{E}}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t(R(s_t,\alpha_t,s_{t+1}) + \alpha F(\pi(.|s_t)))\right] \tag{11}$$

Here, τ denotes the sampled policy trajectory π, $\alpha > 0$ is an entropy term controlled by the relative significance parameter versus reward, which controls, thus, the stochasticity of optimal policy π. In order to guarantee that the predicted amount of rewards is reduced by how far the future rewards are achieved, the γ discount factor is implemented. The best policy is one that maximises the policy's expected return. The expected reward calculation is the summation of discount factors multiplied with the transition reward of state s to state s_{t+1} when the action is α_t, as well as the entropy. In this way, it is possible to describe the value functions that determine the expected return for a certain policy or state action pair.

In the *equation:* 12 the Z_π is the expected return when the starting state s and the given policy π, which incorporates the entropy bonus from each timestep.

$$Z^\pi(s) = \mathop{\mathbb{E}}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t(R(s_t,\alpha_t,s_{t+1}) + \alpha F(\pi(.|s_t)))s_0 = s\right] \tag{12}$$

In the *equation:* 13 the Q^π is the calculation of the expected return when the starting state is s and an arbitrary action α at all the actions of policy π. Furthermore, except for the first timestep, all entropy bonuses will be added.

$$\begin{aligned} Q^\pi(s,\alpha) = \mathop{\mathbb{E}}_{\tau\sim\pi}\Bigg[\sum_{t=0}^{\infty}\gamma^t\Big(R(s_t,\alpha_t,s_{t+1}) \\ +\alpha\sum_{t=1}^{\infty}\gamma^t F(\pi(.|s_t))\Big)\Bigg| s_0 = s, \alpha_0 = (\alpha)\Bigg] \end{aligned} \tag{13}$$

Combining the Z^π and the Q^π from the above two equations,

$$Z^\pi(s) = \mathop{\mathbb{E}}_{\alpha\sim\pi}[Q^\pi(s,\alpha)] + \alpha F(\pi(.|s)) \tag{14}$$

The *equation:* 21 defines the Bellman equation as expressing the value of a decision making task at a given moment in terms of the payoff from some initial option, as well as the value of the remaining decision problem that comes from those initial choices.

$$\begin{aligned} Q^\pi(s,\alpha) &= \mathop{\mathbb{E}}_{\substack{s'\sim p\\ \alpha'\sim\pi}}\Big[R(s,\alpha,s') + \gamma\Big(Q^\pi(s',\alpha') + \alpha F(\pi(.|s'))\Big)\Big] \\ &= \mathop{\mathbb{E}}_{s'\sim p}[R(s,\alpha,s') + \gamma Z^\pi(s')] \end{aligned} \tag{15}$$

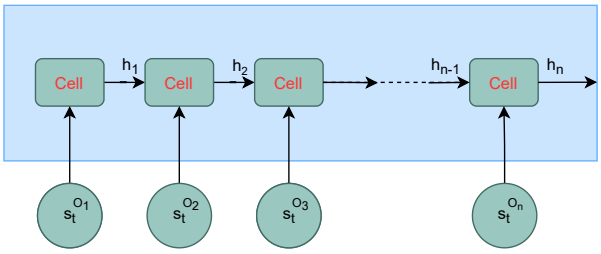


FIGURE 7. *LSTM Module.*

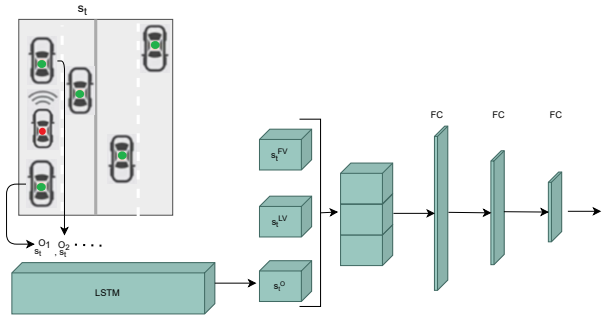


FIGURE 8. *Input Network Structure.*

The following state s' is sampled from the current state transition rules, as indicated by the shorthand $s' \sim P$ for $s' \sim P(.|s,\alpha)$. And the shorthand $\alpha \sim \pi$ is for $\alpha \sim \pi(.|s)$, which denotes that the action was taken based on the policy rules. As a result, the Bellman equation equivalent the summation of both the expected reward of a state change and the value of the state to which the transition is made. Soft actor-critic learns a policy as well as two Q-functions, Q_1 and Q_2, simultaneously. These Q-functions are the approximators for optimum function of action value. With the definition of entropy, the Bellman equation can be rewritten The Bellman equation may be reformulated under the notion of entropy *equation:* 16.

$$Q^\pi(s,\alpha) = \mathop{\mathbb{E}}_{\substack{s'\sim p\\ \alpha'\sim\pi}}\Big[R(s,\alpha,s') + \gamma\Big(Q^\pi(s',\alpha') + \alpha\log\pi(\alpha'|s')\Big)\Big] \tag{16}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access

Since Q^{π} is expected in future states from the replay buffer and the future actions, it could be approximated and rewritten with samples *equation:* 25.

$$Q^{\pi}(s,\alpha) \approx r+\gamma(Q^{\pi}(s',\alpha')-\alpha\log\pi(\tilde{\alpha}'|s')), (\tilde{\alpha}' \sim \pi(.|s') \tag{17}$$

The next actions must be $fresh$ sampled from the policy instead of α', while r and s' are taken from the buffer for the replay. When compared to PPO, the replay buffer should make sure that SAC is always more sample efficient, with more buffer experience being preserved. Sample efficiency implies the agent's expertise to achieve a specific degree of efficiency. In the *equation:* 18 the Mean Squared of Bellman error will reflect how near the Q function is to the Bellman equation. SAC uses a dual-Q method, which is clipped, and between the approximating two Q values is minimum.

$$L(\theta_i,K) = \underset{(s,\alpha,r,s',k)\sim K}{\mathbb{E}}\left[\left(Q_{\theta i}(s,\alpha)-v(r,s',k)\right)^2\right] \tag{18}$$

In *equation:* 19, K is here replay buffer with a target function and k is the finished signal.

$$\begin{aligned} v(r,s',k) = r+\gamma(1-k)(\min_{j=1,2} Q_{\phi t\alpha rgj}(s',\tilde{\alpha'}) \\ -\alpha\log\pi_{\theta}(\tilde{\alpha}'|s')), \quad \tilde{\alpha}'\sim\pi_{\theta}(.|s') \end{aligned} \tag{19}$$

In *equation:* 20 each of the state, in addition to projected future entropy, the policy attempt to maximise expected return and hence maximise Z^{π}.

$$\begin{aligned} Z^{\pi}(s) &= \underset{\alpha\sim\pi}{\mathbb{E}}\left[Q^{\pi}(s,\alpha)\right]+\alpha F(\pi(.|s)) \\ &= \underset{\alpha\sim\pi}{\mathbb{E}}\left[Q^{\pi}(s,\alpha)\right]+\alpha\log\pi(\alpha|s)) \end{aligned} \tag{20}$$

3) DDPG

Typically, Deep Deterministic Policy Gradient (DDPG) [44] utilize DPG-algorithms with the help of Neural Networks as a generic function approximator. Three difficulties arise when Neural Networks (NN) are utilised in RL for any continuous type of action space: instability, insufficient exploration, and correlated data. This chapter is to demonstrate three methodologies to solve these difficulties as a result of the examination of safety efficiency. The neural network training data must be scattered independently and uniformly. When samples are produced in a sequential manner in a simulated environment, this is not always the case. To store collected data as previous experience, the DDPG algorithm uses the replay buffer. When enough data has been accumulated, a replay buffer may be used. The purpose is to avoid the correlation data curse. The squared loss from the samples may then be used to build the loss function for Actor-Critic.

$$\mathcal{L}(\theta^{Q}) = \hat{\mathbb{E}}_{st\sim\rho^{\beta},\alpha_t\sim\beta,r_t\sim E}\left[Q(s_t,\alpha_t|\theta^{Q})-v_t)^2\right] \tag{21}$$

where

$$v_t = r(s_t,a_t)+\gamma Q(s_{t+1},\mu(s_{t+1})|\theta^{Q}) \tag{22}$$

The Critics Neural Network is prone to divergence because v_t is calculated by the same network that is optimising. Making clones of networks and updating them along with gradual upgrades can solve the problem. Making duplicates of both actors and critics has been shown to be the most effective approach for ensuring stability.

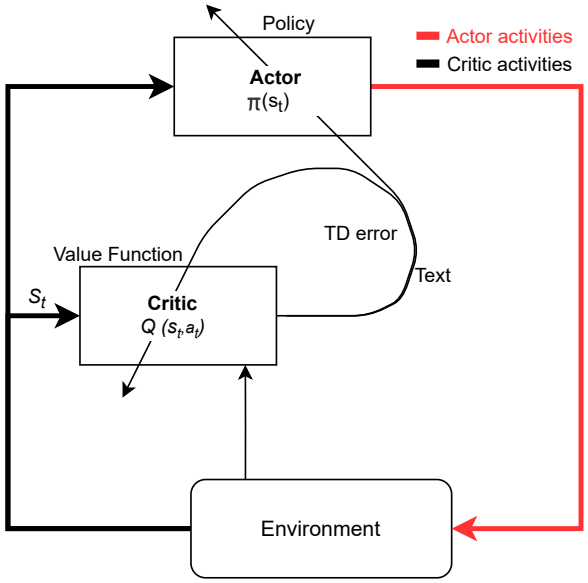


FIGURE 9. *Form of the Actor Critic model network settings.*

The clones are defined as

$$\begin{aligned} Q'(s,\alpha|\theta^{Q}) \\ \mu'(s|\theta^{\mu'}) \end{aligned} \tag{23}$$

and the soft updates can mathematically be formulated as

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^{Q}+(1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^{\mu'}+(1-\tau)\theta^{\mu'} \end{aligned} \tag{24}$$

Here $\tau \ll 1$. Because of the limitless number of possible permutations, exploring continuous action spaces is difficult. In the off-policy techniques, the exploration can be developed independently of the learning mechanism. Incorporating an exploration noise into the actors' action may be the simplest method to generate an exploring actor.

$$u_{exp}(st) = \mu(s_t|\theta_t^{\mu})+P \tag{25}$$

where P is a variable that can be changed depending on the environment.

4) LSTM Structure

The states of the ego vehicle and the other participant vehicles serve as the input to the value network in this experiment. It is particularly unpredictable as to how many moving obstacles (one to four) will collide in the environment, which has a significant effect on policy implementation. It is evident that the driving decision-making process should not consider all states of the environment equally. The manoeuvre policy is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access

affected by the position and velocity of each agent. LSTM is used to collect information from the environment when the quantity of barriers is unknown. LSTM is typically used to manage time series data while also encoding a series of time-independent data. The obstacles are sorted from distant to near in order to generate artificial spatial relations, this means that the most important factor in determining the ultimate hidden state is the barrier that is closest to the narrator. In addition, this type of method reduces the impact of the early states in forgetting state of LSTM. Even with an enormous number of obstacles in its path, the concealed state can handle them all if it is sufficiently big. A sequential input to the LSTM is illustrated in *Figure 7*, where the states information of obstacles is deemed as a serial input to the LSTM, which gets the state information of each barrier one by one and, eventually, outputs an encoded state for all of the barriers. By using this method, we can deal with the problem of multiple obstacles. The value network is created using an LSTM module and the network structure depicted in *Figure 8*. As a final step, we feed this information into three layers of fully connected (FC). As a final step, the value network calculates an estimate of how much the current state is worth. It should be noted that the statuses of participant cars are handled identically in the suggested structure, and no elements of the formation are extracted, which requires the settings of neural network learning how to determine the upkeep quality of the development.

5) Training Network Settings

The speed of the participant vehicles was evenly formed in the range of $[u_i(t)\ , d_m(t)]$ after the beginning state, as well as maintaining a same speed across all of the participating vehicles.The actor network was made up of two hidden layers, that had a total of 64 and 64 units in each and was fully connected. The critic network, on the other hand, was made up of two hidden layers consisting of 64 and 64 + 2 units (the action output: accelerator and steering) in each that are fully connected. *Figure 9* illustrates how actor networks and critic networks work in general.In both the actor network and the critic network, the activation function ReLU (rectified linear unit) was utilized to activate all hidden layers. Because of the action space range, the tanh activation function is used in the output layer of the actor network. The output layer of the critic network, on the other hand, does not have an activation function. A uniform distribution $[-3 \times 10^{-3}, -3 \times 10^{3}]$ was used in both the actor and the critic networks as the weights for the output layer. Actor and critic networks are updated with the Adam optimizer. *Table 1* shows all parameters of the traffic condition and DRL algorithms.

IV. SIMULATION AND SAFETY EFFICIENCY ANALYSIS

In this section, we run the simulations in both single vehicle and multi-agent contexts to validate the effectiveness and accuracy of the anticipated autonomous driving approach. Furthermore, in order to analysis the safety efficiency, we look at how an ego car would interact with the other three participant vehicles as well as with each other in order to train for optimum driving. $TensorFlow$ is used to execute the simulation that implements our deep reinforcement learning methods. To begin, we will go through the simulation scenario creation as well as the parameter settings. Following that, we evaluate the safety efficiency of the techniques by analyzing their uncertainty modeling. According to the $state\ space$ and $action\ space$ description mentioned in *Section III* we have the state space of the environment as, $S = (x_i, y_i, d_{x,i}, d_{y,i}\ i \in 0, ...4_{vehicle})$, and the traffic scenario is made up of a set of vehicle positions (x_i, y_i) and respective speeds $(d_{x,i}, d_{y,i})$ where 0 index corresponds the ego vehicle, while it can see the range $x_b = 20m$,in surrounding.

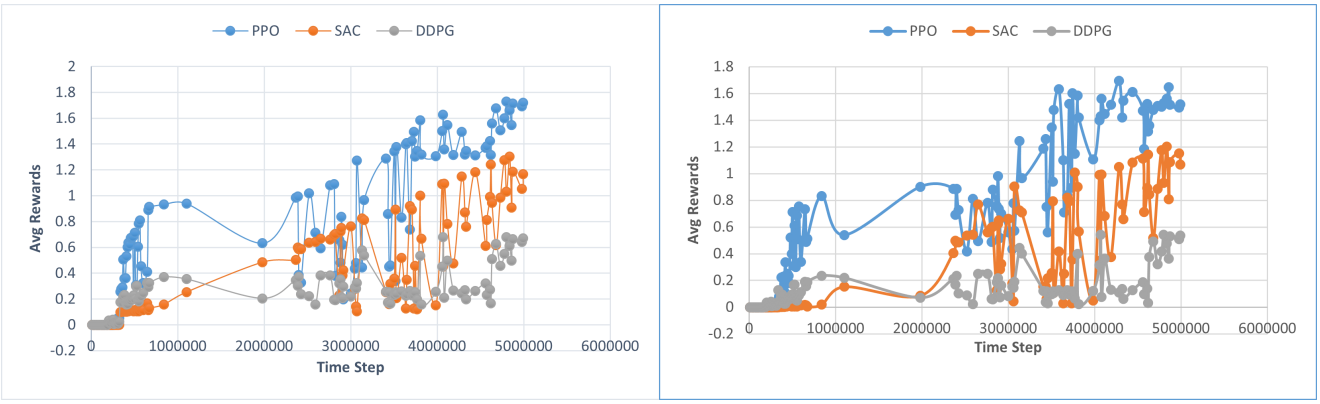
A. AGENT

The created DRL agent has two features: the first is target destination or goal achievement, and the second is partner road user consideration. In this manner, it will be possible to perform effective autonomous braking and avoid the abrupt deceleration and lane change of the leading participant car. For good driving behaviors, a positive reward will be paid to the agent at the end of the episode. It will also be positively rewarded for passing through checkpoints and for its speed. If a speed control fails, the agent will be fined. A small punishment is applied to each decision made by the agent in order to ensure that the most efficient decisions are taken, eventually leading to immediate driving. While modern self-driving automobiles mostly employ effective sensors such as $LiDAR$, this form employs two sets of rays, with eight rays per direction, covering 180 degrees of the entire environment at a $20meters$ length. The first pair of rays determines the distance between the lanes (3.6m in this experiment) and the pavement or roadside infrastructure. The second set of sensors predicts the distance to other cars in the vicinity of the ego vehicle. In addition, the agent vehicle is aware of its own velocity and steering direction, which allows it to drive the vehicle at the appropriate speed. We train two approaches to demonstrate the reliability analysis a single vehicle approach and a multiple agent approach involving $4\ Agents$ per training instance. *Figure:* 9 presents the $RL\ agent$ of our training process.

B. REWARD

The reward function here indirectly establishes the optimization target, which is an important aspect because the reward function is used to characterize optimal behavior. Defining a reward function for driving a vehicle that describes optimal driving behavior. The job is tricky, as there are many different types of driving scenarios that can be hard to derive into an explicit equation. However, because the scope of this study is restricted to the Ego-Vehicle (EV) and four participant cars acting, many of the issues of constructing a driving reward function can be overlooked. We consider a one-way with two-lane scenario in the simulation, as represented in *figure:*2a The vehicle in front of it and the vehicle behind it are in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access



(a) The average rewards of training process of the single agent environment. (b) The average rewards of training process of the multi-agent environment.

FIGURE 10. The average rewards of algorithms in the training process of both single and multi-agent environment.

TABLE 1. *The vehicle dynamics parameter details*

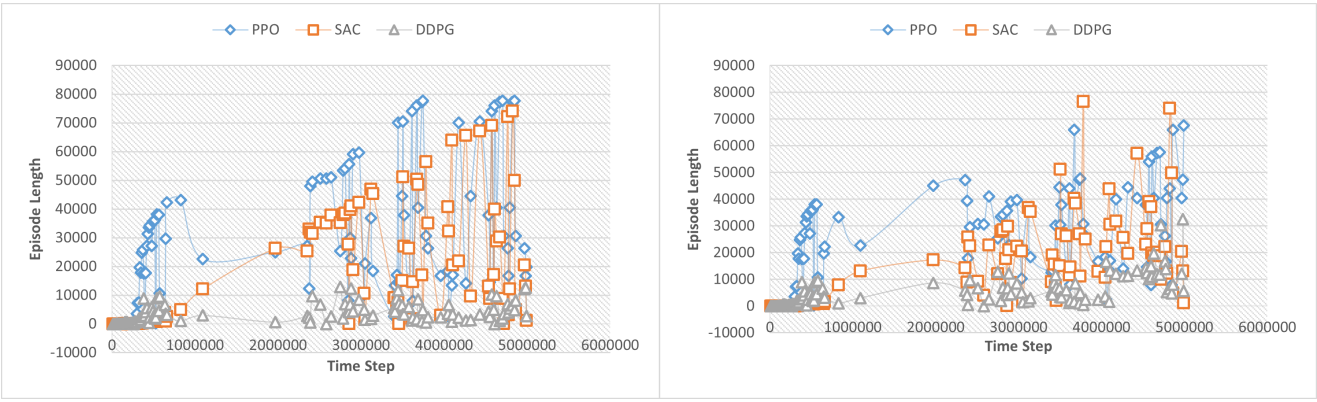
Description	Parameter	Value
Agent vehicle (Ego Vehicle)	EV	$Red\ color\ marked$
Participant vehicles	$FV\ or\ LV$	$Green\ color\ marked$
Vehicle initial speed	$u_i(t)$	$30 \sim 90km/h$
Vehicle maximum speed	$d_m(t)$	$90 \sim 120km/h$
Deceleration rate	$a_p(t)$	$-4 \sim 0m/s^2$
Vehicle normalize steering	$s_{str}(t)$	$-1 \sim 1$
General following distance	$S_{gd}(t)$	7.25m
Therashold distance	$S_{thr}(t)$	4.25m
Ego Vehicle (EV) position	$L_{EV}, (0)$	(0,0)
Flowing Vehicle (FV) initial position	$L_{FV}, (0)$	(0,5)
Other participants vehicle initial position i	$L_i, (0)$	(0,0)
Memory size (Reply)	-	1,000,000
Minibatch size	N	256
Learning rate (actor and critic)	-	0.001
Hidden units (actor network)	-	64,64
Hidden units (critic network)	-	64,64+2=66
Learning rate (actor critic)	-	0.001

the same lane. Randomly, one obstacle will be created in front of the leading vehicle, and the ego vehicle will receive warning from the signal of the taillight of the leading vehicle. Similarly, in *figure:*2b. One Following Vehicle (FV) is driving in the same lane of EV and the Leading Vehicle (LV) moving just right-side lean of the ego lane. Suddenly, the Leading vehicle changes its lane and shifts into the following ego lane. The initial position of the EV and other vehicles is fixed by $Unity\ 3D\ game\ engine$ physics. The initial vehicle speed is chosen at random between $30km/h$ and $90km/h$. The vehicle's deceleration ranges from $-4m/s^2$ to 0, with 0 indicating no braking. The EV, FV and LV have a greater beginning space headway than the safety spacing, and the following distance is equal to $4.25meters$, which is the length of a vehicle. The driving styleLV is chosen at random from the following three scenarios: LV sudden slowdown (the EV, LV, and the FV will be in the same lane or parallel); LV abruptly changes lanes to the EVs front. The LV determines the shifting time at random before changing lanes. LV continues to drive in the same manner as before (the LV and the EV will be in the same lane or parallel). When the distance between the EV and LV goes beyond a threshold value in either of these conditions, the EV chooses a new autonomous control decision to cope with the emergency, namely braking and steering. EV is not required to take any driving action in the third condition. The goal of our agent is to traverse from the starting point to the ending point while staying safe and on schedule. In order to attain this goal, a straightforward and rewarding style is employed. A positive reward based on $2 - \frac{d_{max}-d_{min}}{d_{max}}$ (d indicate the velocity in this study) is received by the agent at each time step, which encourages efficient driving, such as overtaking slow participant vehicles. Alternatively, when a collision happens, or if the EV drives off the road (e.g., if the lane changing happens outside of road boundaries), a penalty or negative reward equivalent to $r_{col} = -2$ is added, and immediately the episode is ended. In addition, the episode does not end when the EV induces another vehicle to emergency brake, which is defined as deceleration with a scale greater than the value for $a_e = 4m/s2$ or when the EV travels closer to another participant vehicles than the temporal gap of $t_{gap} = 2.5seconds$. For further safety and discouraging excessive lane changes, an additional negative reward $r_{lc} = .2$ is given when a lane change is undertaken. *Table:*1 shows the precise parameter settings for vehicle dynamics.

C. SIMULATION SETUP

Unity 3D was used to train our DRL agent vehicle on how to avoid multiple collisions. The Microsoft-developed open-source Unity 3D platform is being utilised to bridge the gap between real-world and computer-simulated autonomous vehicle development. Unity 3D has a lot of advantages. It is a simulator that was made with the Unreal Engine. Besides providing superb visual rendering, the Unreal Engine includes a wealth of capability for collision-related analysis,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access



(a) The length of episode in training process of single-agent environment.

(b) The length of episode in training process of multi-agent environment.

FIGURE 11. The length of episode in training process.

which is very useful. In the process of training an agent through trial and error, it is conceivable for the DRL agent to directly experience collision conditions. The neural network must be fed with data from the developed environment in order to be properly trained. Driving the agent vehicle and the participant vehicle in Unity 3D allows us to acquire the data that is required for learning. Different types of vehicle sensors can be used with Unity 3D. In particular, the LiDER and the Inertial Navigation System (INS) are equipment on agent and participant vehicles. Using these sensors, we can assemble important data about the current state of the vehicle.

The reinforcement learner may successfully regulate the vehicle speed of a trained agent in a multi-agent collision avoidance environment using either DDPG or any other method like SAC and PPO, thanks to an extensive simulation designed to handle the problem of unexpected slowing and lane shift. The ML-Agents toolkit from Unity is utilised to assist the neural network, which uses the open-source TensorFlow library to construct and form machine learning models. In this 3D simulation, the car can be subjected to gravitational, rolling, and dredging impacts. The SAC algorithm's training process is seen in *Figure:*9. The driving safety effectiveness concerning the evaluation of chain collision avoidance techniques will be determined in this work, which has two dimensions: single agent and multiple agent environments.

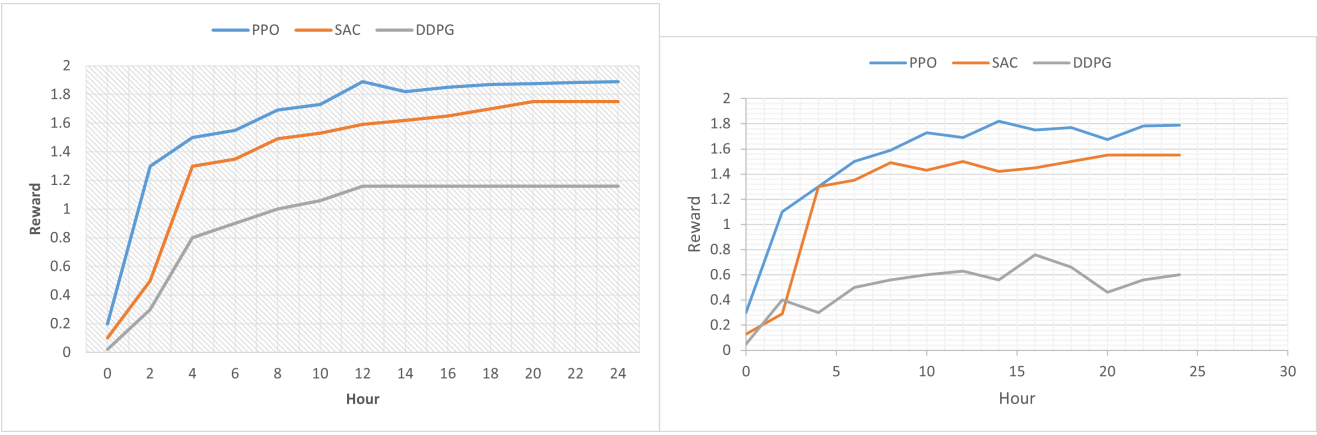
D. RESULTS

In this section, episode length, average reward, average inference rewards, value loss, and the number of collisions per episode, in particular episodes during training, are compared to the driving efficiency and overall safety efficiency of the three algorithms. The agents are required to execute 5 million episodes in order to evaluate the three algorithms in the context of two distinct aspects (single-agent and multiple-agent environments). Overall safety efficiency is determined by the agents' total success rate. To quantify efficiency and assess the dependability of these three methods, the total amount of guidance and the average timesteps per episode are employed. The rest of the time, each vehicle on the road has its own initial position and travels at velocities ranging from 90 to 120 kilometres per hour. All vehicles have sensors attached to them to perceive the existence of other vehicles. If it becomes vacant, it is a chance to slow down and operate the vehicle braking system at random. The agent vehicle will slow down if there is a car in front of it (Leading Vehicle (LV)). When a vehicle approaches from behind, it accelerates. When a vehicle is in the front and back, it ensures that the same spacing between the vehicles is maintained. We assess the safety effectiveness by examining the four sets of figures offered, each of which depicts two different approaches to our training performance. First, the *Figure:*10 presents average rewards with time steps, an illustration of the learning efficiency of our modelled agent in both single and multi-vehicle environments. Second, the *Figure:*11 shows the episode length with time steps, and it gives valuable insights about how many unusual actions they take and the number of accidents they have per episode. Third, the *Figure:*12 reveals the inference average rewards that indicate the overall collision avoidance performance and driving smoothness of the trained agent. The policy loss is depicted in the fourth *Figure:*13. From this figure, we can determine the overall training perspective as to whether the agent can learn optimum behaviour or not. And finally, the consecutive two figures of *Figure:*14 present the particular time step collision numbers per episode.

E. UNCERTAINTY MODELING ANALYSIS

We initially train our chosen algorithms, PPO, SAC, and DDPG, to model the uncertainty in traffic circumstances for both single-agent and multi-agent autonomous traffic flow. The following table shows the training data *table:*2 provides us with useful information for making driving decisions, such as vehicle speed and the relative distance between a vehicle and an item in front of it. According to our proposed driving strategy depicted in both equations *equation:*1, and *equa-*

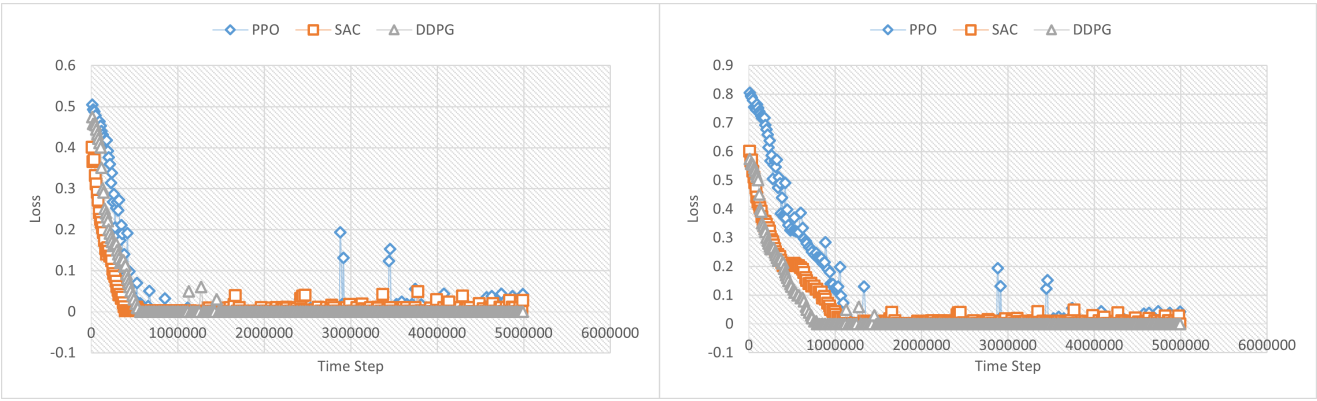
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access



(a) The average inference rewards of the single agent environment.

(b) The average inference rewards of the multi-agent environment.

FIGURE 12. The average inference rewards both single and multiple agent environments.



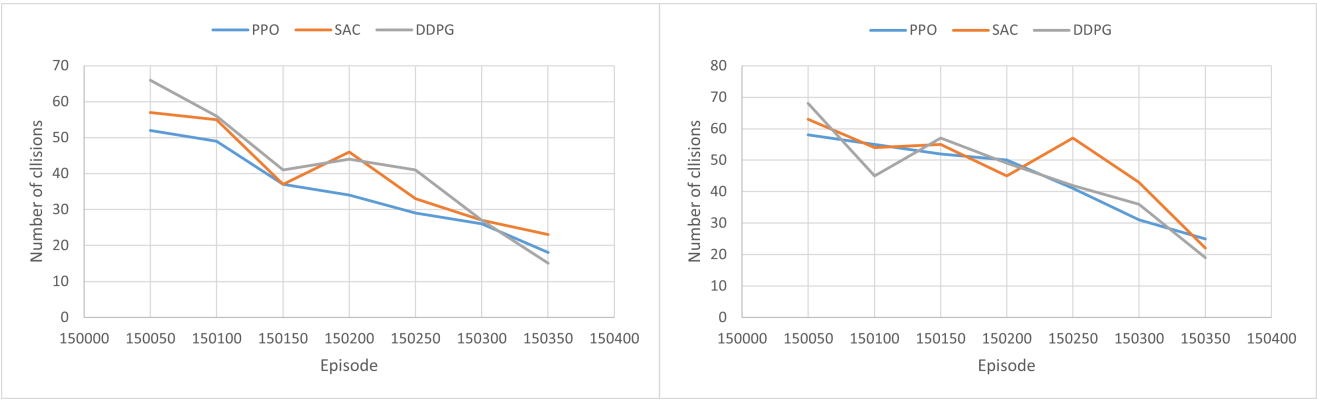
(a) The value loss presentation of single agent training.

(b) The value loss presentation of multi-agent training.

FIGURE 13. The value loss estimation of both single and multi-agent environment

*tion:*2, it contains both a single and multi-agent environment, and the training process has been done over a lengthy time period (24 hours) with a single and multiple vehicles (here it is 4 vehicles). It is possible that it does not account for all of the sources of uncertainty in the simulated driving environment. The results reported in these figures and tables are assumed to be reflective of the uncertainty sources in the driving environment of single and multi-vehicle traffic flow described in this article. Other sources of uncertainty, such as weather, road conditions, and road profile, should be incorporated into the future safety efficiency study of autonomous vehicle driving decision making. *Table:*2 depicts two aspects of a single agent environment and a multi-agent environment for training scenarios in which the three algorithms perform differently. The agent can control the vehicle's velocity on the road with an 88 percent success rate. When vehicles are close together, the agent either decelerates or accelerates based on the distance to the other participants. The three DRL algorithms repeated the process for a total of $5,000,000$ times or for a total of 24 hours of training. Because the episode's highest reward is 88.2, an average of 55 is considered good. While PPO required less time than SAC, the DDPG never reached this average value (for the multiple agent environment). The average reasonable reward is .6, in which PPO took roughly 2 million (single agent) and 3.1 million (multi agent) timesteps compared to the SAC, 3.1 (single agent) steps, 4.2 (multi agent) steps, and 4.7 million timesteps for single agent and not applicable (for multi-agent) of the DDPG method. As indicated by a quick drop in performance at 1.5 and 1.8 million timesteps, the amount of time of PPO increase was more than stable DDPG and SAC, and its performance swings remained reduced. A superior success rate has been attained. The amount of time each episode appears is the same for all three methods. The DDPG and SAC agents, on the other hand, had approximately twice as much steering each episode. The SAC agent approximately doubles the number of steering fractions per episode, despite the fact that the timesteps for the three algorithms are the same each episode, implying that all agents execute the episode with the same number of actions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3167812, IEEE Access



(a) Number of collision in per episode in particular time steps of single-agent environment.

(b) Number of collision in per episode in particular time steps of multi-agent environment.

FIGURE 14. Number of collision in per episode in particular time steps.

TABLE 2. *The overall Reliability Analysis of three RL Algorithms in the aspect of single and multi-agent environments.*

Algorithm	**PPO**		**SAC**		**DDPG**	
Training Approach	**Single Agent**	**Multi-Agent**	**Single Agent**	**Multi-Agent**	**Single Agent**	**Multi-Agent**
Success rate	88%	81%	67%	61%	35%	29%
Timesteps till .6 rewards	2 mill	3.1 mill	4.1 mill	4.2 mill	4.6 mill	N/A (within 5 mill)
Timesteps	3 mill	4.1 mill	4.2 mill	4.6 mill	5 mill	5 mill
Time till .6 rewards	8 hours	13 hours	8 hours	15 hours	24 hours	24 hours
Training hours	10	15	12	18	24 hours	24 hours
Per episode avg steering change	54	83	177	197	93	105
Per episode avg timesteps	61	86	72	89	165	192

F. SAFETY EFFICIENCY ANALYSIS

We then do a collision-avoidance safety efficiency analysis utilising the training results stated in the previous subsections, centred on the proposed driving technique for quantifying the uncertainty in chain collision avoidance. The training process deploys chain collision avoidance strategies in autonomous driving and analyses the safety efficiency according to the aspect of chain collision avoidance driving manoeuvres with respect to the three RL algorithms (DDPG, PPO, and SAC) by allowing an agent to learn how to do so.

- The training results show that the recommended driving approach is effective, and they point us in the right direction to enhance the accuracy and efficiency of the collision-avoidance trustworthiness analysis. *Figure:2* shows the collision risk and avoidance failure prediction as a function of the controller parameters velocity v and time t. We look at the training results in both single-agent and multi-agent scenarios to see how well or poorly they perform. In terms of overall performance, the PPO, SAC, and DDPG efficiency evaluations are all equal, with success rates of 88, 81, and 67 percent for single vehicle environments, while 61, 35, and 29 percent for multiple vehicle environments, respectively. In terms of learning rates, the PPO algorithm needed less time and less episodes than the SAC and DDPG to reach this rate of success, indicating sample efficiency. Here we can primarily see that the PPO is more reliable than the other two algorithms, SAC and DDPG.
- On the other hand, DDPG and SAC took longer to process 5 million episodes than PPO, owing to the fact that both DDPG and SAC are off-policy updates. Because it learns from all previous experiences, the DDPG or SAC over PPO should be used when the sample supply is low. PPO trains more quickly, but it requires more data to be successful. When it comes to driving efficiency, the DDPG and SAC agents used more steering in total and took longer timesteps to complete an episode. The algorithm PPO took 54 and 83 timesteps to accomplish speed control, DDPG took 93 and 105, and SAC took 177 and 197 timesteps, demonstrating that SAC conducted more inefficient actions than PPO and DDPG. It implies that as timesteps are smaller, the likelihood of chain collision failure decreases. However, increasing the number of timesteps may increase the frequency of unneeded braking and steering motions, thereby causing discomfort for the passenger and increasing fuel usage.
- There was also a significant disparity in the amount of total direction done by the agents. Although PPO used more data than the other two, both the DDPG and SAC agents committed more unneeded and worse actions on average, making them sub-optimal in comparison to PPO. However, like with most machine learning

tasks, the problem of overfitting occurs with all DDPG, SAC, and PPO. The performance of the three algorithms is shown in *table:*2. Based on this table information, we can derive an adequate measurement of the safety efficiency of our three algorithms, which is that PPO is more trustworthy than SAC and DDPG, and the DDPG performs poorly in both phases of the training period (i.e., single-agent and multi-agent).

- According to *Figure:14*, we can evaluate the safety efficiency in both single vehicle and multiple vehicle environments. We clearly see that there is no significant difference between a multiple vehicle environment and a single vehicle environment. Two figures show that the algorithms' performances are more or less the same. More precisely, the PPO algorithm gives us a smaller number of collisions at the start of the episode at 150050. However, the DDPG gives us a lower number of collisions at the end of this episode range as 150300-150350, which is 15 (for single-agent) and around 20 (for single-agent). So, for this, we can say that the DDPG and SAC also have good perspectives for avoiding collisions. But the overall performance of PPO steadily decreases from start to ending episode in this particular range.

Based on the foregoing data, performing a safety efficiency study accurately requires a particular number of training sessions regardless of the method of measurement assessment used. More training will allow us to focus even more on small regions that are crucial for safety analysis, allowing us to improve the efficiency of collision-avoidance decision-making reliability analysis.

G. COMPARISON OF COMPUTATIONAL COMPLEXITY AND EFFICIENCY

The computational complexity of the proposed method can be stated as: $O(\Psi)$,

where, $\Psi = OH + OMY + HI + MYI$

i.e.

$$O(\Psi) = O(GH + GMY + HI + MYI) \qquad (26)$$

Where G represents the number of output units, H represents the number of hidden units, M represents the number of memory cell blocks used, Y represents the size of memory cell blocks used, and I represents the maximum number of forward-connected units of memory cells, hidden units, and gate units, as well as the number of weights. According to the structure of deployed training networks, it can be said that, overall, the techniques (PPO, SAC, and DDPG) are local in time and space, which means that the values of activation obtained during the sequence processing phase do not need to be stored or kept. Furthermore, the amount of storage it requires is independent of the length of the sequence input.

On the other hand, the simulations were focused on the avoidance of chain collisions among multiple autonomous vehicles. Because of that, we conducted three well-established RL algorithms and tried to investigate their performance according to our problem. In this way, we utilize the off-policy methods as DDPG, and we know that the DDPG algorithm learns a Q-function and a policy at the same time. DDPG also performs "soft updates" (also known as "conservative policy iterations") on both actor and critic networks. In addition, we used SAC, an off-policy method that combines off-policy updates with a stable stochastic actor-critic scheme. To avoid chain collisions, we deployed it because the policy can learn multiple means of optimal behavior. On the other hand, we studied the on-policy updating method as PPO, in which the agent interacts directly with the environment, learns and discards a batch of experiences after performing a gradient update. In the following sections, we will compare the computational cost of our proposed method in single agent and multiple agent systems.

1) Single Agent Environment

According to the simulation results mentioned in *table:*2, it is clearly seen that in the aspect of single agent environment, the required training time for reaching considering reward .6 are for PPO and SAC, which are 8 hours and 8 hours, whereas the DDPG algorithm took 24 hours but did not reach this level of reward. So, for these, we can say that the PPO and SAC algorithms are more cost-effective than the DDPG algorithm. Another matter is the timesteps: they trained the model for 5 million timesteps, and in this time duration, the DDPG algorithm did not complete its training process, which means it took total timesteps and did not achieve the expected rewards. whereas PPO took 2 million and SAC took 4.1 million timesteps to train the model perfectly. Hence, we can consider the PPO to be more cost-efficient than the SAC and DDPG algorithms.

2) Multi-Agent Environment

In this experiment, we first investigate the single vehicle environment to evaluate the cost efficiency of both single and multi-agent driving environments. In the aspect of multiple vehicle environments, every algorithm faced some curtails suffering. If we go for the required training time, the PPO takes 13 hours, and the SAC takes 15 hours. At the same time, the DDPG algorithm took 24 hours. According to our reward function, all the algorithms took almost two times more time to train the model. So, we can decide that the computational costs are higher in multi-agent systems than in single-vehicle environments. And the PPO is more cost-effective than the other two algorithms (SAC and DDPG). In the second aspect, which is time steps, to collect the reward. At .6, all algorithms took more time steps compared to the single vehicle environment. More specifically, PPO took 4.1 million, SAC took 4.2 million, and DDPG took 5 million. Here also, the computational cost of DDPG is higher than PPO and SAC.

V. CONCLUSION

This study presented a deep reinforcement learning-based driving technique for avoiding multiple vehicle collisions. Using these strategies, we suggested the challenge of obtaining both ego and participant vehicle state information via sensors. Based on these considerations, we built a Markov decision process to describe the collision avoidance technique and a reward function for collision avoidance, as well as integrate the agent vehicle with the PPO, SAC, and DDPG algorithms. The results of our study show that the agent vehicle effectively performed the avoidance of multiple-vehicle collisions.

Various adverse weather circumstances and the road profile are also possible sources of uncertainty that need to be examined more thoroughly. This should be studied further as part of our research. This simulation, in actuality, simplifies things too much. Although tasks using computer vision are vital for estimating the surroundings of an autonomous vehicle, they are mostly ignored in this simulation. Future simulation studies should integrate these conditions to simulate more realistic road situations.

ACKNOWLEDGMENT

This research is supported by Ministry of Higher Education of Malaysia under Fundamental Research Grant Scheme (FRGS/1/2018/TK08/UMP/02/2). We gratefully acknowledged Adam Streck's contribution. For our experiment, we used his Unity environment. The GitHub repo can be found [GitHub](https://github.com/xstreck1/cAr-drIve).

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] T. Dirsehan and C. Can, "Examination of trust and sustainability concerns in autonomous vehicle adoption," *Technology in Society*, vol. 63, p. 101361, 2020.

[2] S. Y. Tan and A. Taeihagh, "Adaptive governance of autonomous vehicles: Accelerating the adoption of disruptive technologies in singapore," *Government Information Quarterly*, vol. 38, no. 2, p. 101546, 2021.

[3] J. Barnett, N. Gizinski, E. Mondragon-Parra, J. E. Siegel, D. Morris, T. Gates, E. Kassens-Noor, and P. Savolainen, "Automated vehicles sharing the road: Surveying detection and localization of pedalcyclists," *IEEE Transactions on Intelligent Vehicles*, 2020.

[4] K. Othman, "Public acceptance and perception of autonomous vehicles: a comprehensive review," *AI and Ethics*, pp. 1–33, 2021.

[5] J. Dahl, G. R. de Campos, C. Olsson, and J. Fredriksson, "Collision avoidance: A literature review on threat-assessment techniques," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 101–113, 2019.

[6] X. Wang, X. Zheng, W. Chen, and F.-Y. Wang, "Visual human-computer interactions for intelligent vehicles and intelligent transportation systems: The state of the art and future directions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[7] M. T. Hussain, N. B. Sulaiman, M. S. Hussain, and M. Jabir, "Optimal management strategies to solve issues of grid having electric vehicles (ev): A review," *Journal of Energy Storage*, p. 102114, 2020.

[8] J. Barnett, N. Gizinski, E. Mondragon-Parra, J. E. Siegel, D. Morris, T. Gates, E. Kassens-Noor, and P. Savolainen, "Automated vehicles sharing the road: Surveying detection and localization of pedalcyclists," *IEEE Transactions on Intelligent Vehicles*, 2020.

[9] R. Mounce and J. D. Nelson, "On the potential for one-way electric vehicle car-sharing in future mobility systems," *Transportation Research Part A: Policy and Practice*, vol. 120, pp. 17–30, 2019.

[10] S. Thormann, A. Schirrer, and S. Jakubek, "Safe and efficient cooperative platooning," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[11] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[12] T. Peng, L. Su, R. Zhang, Z. Guan, H. Zhao, Z. Qiu, C. Zong, and H. Xu, "A new safe lane-change trajectory model and collision avoidance control method for automatic driving vehicles," *Expert Systems with Applications*, vol. 141, p. 112953, 2020.

[13] L. C. Kiew, A. J. M. Muzahid, and S. F. Kamarulzaman, "Vehicle route tracking system based on vehicle registration number recognition using template matching algorithm," in *2021 International Conference on Software Engineering Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, 2021, pp. 249–254.

[14] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[15] B. Hirchoua, B. Ouhbi, and B. Frikh, "Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy," *Expert Systems with Applications*, vol. 170, p. 114553, 2021.

[16] M. Rahim, M. Rahman, M. A. Rahman, A. J. M. Muzahid, S. F. Kamarulzaman *et al.*, "A framework of iot-enabled vehicular noise intensity monitoring system for smart city," in *International Conference on Innovative Technology, Engineering and Science*. Springer, 2020, pp. 194–205.

[17] A. M. Boggs, R. Arvin, and A. J. Khattak, "Exploring the who, what, when, where, and why of automated vehicle disengagements," *Accident Analysis & Prevention*, vol. 136, p. 105406, 2020.

[18] N. Sugiyama and T. Nagatani, "Multiple-vehicle collision induced by a sudden stop in traffic flow," *Physics Letters A*, vol. 376, no. 22, pp. 1803–1806, 2012.

[19] Z. Li and L. Chen, "Effects of intelligent control mechanism on multiple-vehicle collision under emergency," *Physica A: Statistical Mechanics and its Applications*, vol. 404, pp. 16–25, 2014.

[20] T. Nagatani, "Effect of velocity-dependent friction on multiple-vehicle collisions in traffic flow," *Physica A: Statistical Mechanics and its Applications*, vol. 465, pp. 636–643, 2017.

[21] A. J. M. Muzahid, S. F. Kamarulzaman, and M. A. Rahim, "Learning-based conceptual framework for threat assessment of multiple vehicle collision in autonomous driving," in *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*. IEEE, 2020, pp. 1–6.

[22] A. Sinha, S. Chand, K. P. Wijayaratna, N. Virdi, and V. Dixit, "Comprehensive safety assessment in mixed fleets with connected and automated vehicles: A crash severity and rate evaluation of conventional vehicles," *Accident Analysis & Prevention*, vol. 142, p. 105567, 2020.

[23] J. Wang, Y. Zheng, Q. Xu, J. Wang, and K. Li, "Controllability analysis and optimal control of mixed traffic flow with human-driven and autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[24] J. Wang, Y. Zheng, Q. Xu, and Wang, "Controllability analysis and optimal control of mixed traffic flow with human-driven and autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[25] S. V. Albrecht and P. Stone, "Autonomous agents modelling other agents: A comprehensive survey and open problems," *Artificial Intelligence*, vol. 258, pp. 66–95, 2018.

[26] G. A. de Morais, L. B. Marcos, J. N. A. Bueno, N. F. de Resende, M. H. Terra, and V. Grassi Jr, "Vision-based robust control framework based on deep reinforcement learning applied to autonomous ground vehicles," *Control Engineering Practice*, vol. 104, p. 104630, 2020.

[27] A. J. M. Muzahid, M. A. Rahim, S. A. Murad, S. F. Kamarulzaman, and M. A. Rahman, "Optimal safety planning and driving decision-making for multiple autonomous vehicles: A learning based approach," in *2021 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, 2021, pp. 1–6.

[28] Y. Zhuang, Z. Pu, J. Hu, and Y. Wang, "Illumination and temperature-aware multispectral networks for edge-computing-enabled pedestrian detection," *IEEE Transactions on Network Science and Engineering*, 2021.

[29] Q. Chen, Y. Xie, S. Guo, J. Bai, and Q. Shu, "Sensing system of environmental perception technologies for driverless vehicle: A review of state of the art and challenges," *Sensors and Actuators A: Physical*, vol. 319, p. 112566, 2021.

[30] N. Sugiyama and T. Nagatani, "Multiple-vehicle collision in traffic flow by a sudden slowdown," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 8, pp. 1848–1857, 2013.

[31] X.-Y. Lu and J. K. Hedrick, "Practical string stability for longitudinal control of automated vehicles," *Vehicle system dynamics*, vol. 41, pp. 577–586, 2004.

[32] N. Kauffmann, F. Winkler, F. Naujoks, and M. Vollrath, ""what makes a cooperative driver?" identifying parameters of implicit and explicit forms of communication in a lane change scenario," *Transportation research part F: traffic psychology and behaviour*, vol. 58, pp. 1031–1042, 2018.

[33] N. Arafat, M. I. Pramanik, A. J. M. Muzahid, B. Lu, S. Jahan, and S. A. Murad, "A conceptual anonymity model to ensure privacy for sensitive network data," in *2021 Emerging Technology in Computing, Communication and Electronics (ETCCE)*. IEEE, 2021, pp. 1–7.

[34] R. Naja *et al.*, *Wireless vehicular networks for car collision avoidance*. Springer, 2013, vol. 2013.

[35] S. S. Kamble, A. Belhadi, A. Gunasekaran, L. Ganapathy, and S. Verma, "A large multi-group decision-making technique for prioritizing the big data-driven circular economy practices in the automobile component manufacturing industry," *Technological Forecasting and Social Change*, vol. 165, p. 120567, 2021.

[36] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[37] T. Kessler and A. Knoll, "Cooperative multi-vehicle behavior coordination for autonomous driving," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1953–1960.

[38] P. Vyas, L. Vachhani, and K. Sridharan, "Interval analysis technique for versatile and parallel multi-agent collision detection and avoidance," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 3, pp. 705–720, 2020.

[39] S. C. Nagavarapu, L. Vachhani, and A. Sinha, "Multi-robot graph exploration and map building with collision avoidance: A decentralized approach," *Journal of Intelligent & Robotic Systems*, vol. 83, no. 3, pp. 503–523, 2016.

[40] D. Spensieri, J. S. Carlson, F. Ekstedt, and R. Bohlin, "An iterative approach for collision free routing and scheduling in multirobot stations," *IEEE Transactions on Automation science and Engineering*, vol. 13, no. 2, pp. 950–962, 2015.

[41] H. Yang, Z. Wang, K. Xie, K. Ozbay, and M. Imprialou, "Methodological evolution and frontiers of identifying, modeling and preventing secondary crashes on highways," *Accident Analysis & Prevention*, vol. 117, pp. 40–54, 2018.

[42] U. Z. A. Hamid, M. H. M. Ariff, H. Zamzuri, Y. Saito, M. A. Zakaria, M. A. A. Rahman, and P. Raksincharoensak, "Piecewise trajectory replanner for highway collision avoidance systems with safe-distance based threat assessment strategy and nonlinear model predictive control," *Journal of Intelligent & Robotic Systems*, vol. 90, no. 3, pp. 363–385, 2018.

[43] A. J. M. Muzahid, S. F. Kamarulzaman, and M. A. Rahman, "Comparison of ppo and sac algorithms towards decision making strategies for collision avoidance among multiple autonomous vehicles," in *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*. IEEE, 2021, pp. 200–205.

[44] T. A. Ribeiro, "Deep reinforcement learning for robot navigation systems," Ph.D. dissertation, 2019.

ABU JAFAR MD MUZAHID received the B.S. (Hons.) and the M.S. degree in Statistics from the Shahjalal University of Science and Technology, Sylhet, Bangladesh. He is currently pursuing his second Master's degree in Computer Systems and Software Engineering at Universiti Malaysia Pahang, Malaysia. His primary research interests are artificial intelligence, machine learning, and green automotive technology.

SYAFIQ FAUZI KAMARULZAMAN received the B.Eng. degree from University of Tsukuba, Japan in 2010. He received his Master and Ph.D. degrees from University of Tsukuba, Japan in 2012 and 2015 respectively. He was a fellow of Universiti Malaysia Pahang (2012-2015), and currently he is a senior lecturer in Faculty of Computing, Universiti Malaysia Pahang. His interest in research focuses in Control Systems and Artificial Intelligence, which includes applications on Autonomous Vehicles, Robotics, and Internet of Things. He is also a member of IEEE since 2010, a member of Malaysian Robotics and Automation Society, a certified Cisco Instructor for CCNA program and Internet of Things.

MD ARAFATUR RAHMAN received his Ph.D. degree in Electronic and Telecommunications Engineering from the University of Naples Federico II, Naples, Italy in 2013. He has more than 10 years Research and Teaching experience in the domain of Computer and Communications Engineering. Currently, he is a Senior Lecturer with the School of Mathematics and Computer Science, University of Wolverhampton, UK. He was an Associate Professor with the Faculty of Computing, Universiti Malaysia Pahang, where he had conducted Undergraduate and Masters Courses and supervised more than 21 B.Sc., 5 M.Sc. and 5 PhD students. He worked as a Postdoctoral Research Fellow with University of Naples Federico II in 2014 and Visiting Researcher with the Sapienza University of Rome in 2016. His research interests include Internet-of-Things (IoT), Wireless Communication Networks, Cognitive Radio Network, 5G, Vehicular Communication, Cyber Physical System, Big Data, Cloud-Fog-Edge Computing, Machine Learning and Security. He has developed an excellent track record of academic leadership as well as management and execution of international ICT projects that are supported by agencies in Italy, EU and Malaysia. Dr. Rahman has received number of prestigious international research awards, notably the Best Paper Award at ICNS-2015 (Italy), IC0902 Grant (France), Italian Government PhD Research Scholarship and IIUM Best Masters Student Award, Best Supervisor Award at UMP, Awards in International Exhibitions including Best Innovation Award, MTE-2020, Malaysia, Diamond and Gold in BiS-2017 UK, Best of the Best Innovation Award and Most Commercial IT Innovation Award, Malaysia, and Gold and Silver medals in iENA-2017 Germany. Dr. Rahman has co-authored of around 100 prestigious IEEE and Elsevier journal (e.g., IEEE TII, IEEE TSC, IEEE COMMAG, Elsevier JNCA, Elsevier FGCS etc.) and conference publications (e.g., IEEE Globecom, IEEE DASC, etc.) and has served as an Advisory Board Member, Editor (Computers, MDPI), Lead Guest Editor (IEEE ACCESS, Computers), Associate Editor (IEEE ACCESS), Petron, General Chair, Organizing Committee, Publicity Chair, Session Chair, Programme Committee and Member of Technical Programme Committee (TPC) in numerous leading conferences worldwide (e.g., IEEE Globecom, IEEE DASC, IEEE iSCI, IEEE ETCCE etc.) and Journals. His name was enlisted inside the World top 2% scientists list released by Stanford University under the category of "Citation Impact in Single Calendar Year 2019". He was a Fellow of IBM Center of Excellence and Earth Resources & Sustainability Center, Malaysia and a Senior Member of IEEE.

ALI H ALENEZI received the B.S. and M.S. degrees in Electrical Engineering from King Saud University, KSA, and Royal Institute of Technology KTH, Sweden, respectively, and the Ph.D. degree in Electrical Engineering from New Jersey institute of Technology, USA, in 2018. He is currently with Northern Border University, Saudi Arabia, as Assistant Professor in the Electrical Engineering Department. His research interests include acoustic communication, wireless communications, and 4G and 5G networks using UAVs.