Reinforcement Learning for Switching Control of Semi-automated Vehicles with Driver Fatigue

Irem Ugurlu

June 27, 2021

Abstract

Automated driving is a rapidly growing technology nowadays. Semi-automated driving is a subpart of automated driving which has multiple driving modes where both driver and automated module can take control. But full safety and comfort guarantees cannot still be given to the drivers. In this project, research has been done to ensure driver safety and comfort for the driver on the decision logic module of a semi-automated vehicle. Research focuses on just one specific use-case which is driver fatigue. The main goal is to ensure driver safety and comfort in the case where the user is sleepy during manual driving. Markov Decision Process is used to create a model to successfully represent this case. Evaluation has been done using already existing Reinforcement Learning algorithms and comparing their performances with the decision-tree based baseline policy. In conclusion, a successful Markov Decision Process model is created. While evaluating, some models performed better than expected and some are worse. In the end, a successful model is proposed, it can still be developed further to provide more safety and comfort for the driver.

1 Introduction

Automated vehicle technology is developing quickly for all transport modes, by providing huge safety capacity. Even though the automated system may not work well in all situations and the human driver needs to take over the control when it is required. This project is built on top of the MEDIATOR project which aims to develop a mediating system for drivers in automated vehicles [1]. The purpose of the MEDIATOR project is to provide safety and comfort for the driver while switching between driver and automated system based on who is the fittest to drive at the moment. The MEDIATOR system consists of five modules, namely: the automation module, the context module, the driver module, the decision logic module, and the human-machine interface (HMI) module. This research focuses on the decision logic module. The decision logic module decides the optimal actions regarding the switching between the driver and the automated system to guarantee both safety and driver comfort in various situations.

The MEDIATOR project focuses on semi-automated vehicles. Semi-automated vehicles have both the manual and the automation mode. Research has been done in semi-automated vehicles in the past, with an optimal switching policy between driver and automated systems being proposed [2]. The results of prior research are useful because these results indicate

that an effective optimal switching policy can be created in semi-automated vehicles. It uses Markov Decision Process (MDP) to determine the optimal switching policy. MDP provides a mathematical framework for modeling decision-making problems in various complex situations. Also, another research has been done which focuses on creating an MDP model for the decision-making problems of the MEDIATOR in specific. This work of Vermunt [3] gives relevant background information about MEDIATOR and designs an MDP model for the decision logic part of the MEDIATOR project. Research of Vermunt is useful for this research since it indicates that the application of MDPs to the MEDIATOR decision logic module is likely feasible and effective. But it does not guarantee safety and driver comfort in all cases.

Cases, where the driver safety and comfort need to be guaranteed can be split into several parts since there exist many cases such as distraction, fatigue, uncomfortable events, sensor failures, etc... This research focuses on the case of fatigue. Prior research about the effect of fatigue in autonomous driving indicates that fatigue in drivers leads to worsened decision making, slower reaction times, reduced attention to the forward roadway, and driving performance incapability [4–8]. Also, it is given that sleepiness and fatigue are contributing factors in 5-50% of all crashes [9]. Thus, this research aims to guarantee driver safety and comfort in the case of driver unfitness because of fatigue by creating an MDP model to determine optimal actions within the decision logic module. This model will be created and then will be evaluated by implementing existing reinforcement algorithms.

The main research question which is aimed to answer in this work is: **How can we create an MDP model that allows the mediator to determine to correct fatigue, shift to automation mode, or emergency stop when the driver becomes unfit because of fatigue?**

In conclusion, this research will try to find the most appropriate way to model fatigue as an MDP to allow the mediator to do the correct action and provide safety and comfort to the driver. In order to achieve this, some more background information will be given in section 2. Section 3 will follow this with an explanation of the methodology of this research including the MDP formulation. Results and evaluation will be explained in section 4. Section 5 will be focused on discussion. Section 6 is the responsible research section. And finally, section 7 will talk about the conclusion.

2 Background

In this section of the paper, detailed background information about the used techniques such as MDP and Reinforcement Learning (RL) will be given. Also why the usage of these techniques is chosen will be explained.

2.1 Markov Decision Process

MDP provides a mathematical framework for modeling decision-making problems in various complex situations. MDP is used in this research since previous researches [2, 3] indicate that it is a successful technique to model decision-making problems in autonomous driving. A Markov Decision Process consists of 4 parts. Namely, state-space, action space, transition probabilities, and reward function.

- **State Space** consists of the states which are the possible situations of the agent at a specific time-step in the environment. So, whenever the agent performs an action the agent ends up in another new possible state from the state space. In this research, these are the possible levels of the automation mode.
- **Action Space** consists of the different possible actions the decision logic module can take.
- **Transition Probabilities** are the probabilities of an agent to go from one state to another state by performing a specific action.
- **Reward Function** represents the reward the agent can get by performing a specific action in a specific state.

2.2 Reinforcement Learning

RL is one of the sections of machine learning, in which the system learns from its own actions. In this research, RL is chosen to evaluate the created MDP model and these results will also be compared with the baseline decision tree-based policy. It is expected that RL algorithms will perform better than baseline decision tree-based policy. Following Reinforcement Learning methods are used during the evaluation part of this research.

- **Deep-Q Learning (DQN):** A reinforcement learning algorithm that combines Q-Learning with deep neural networks to let RL work for complex, high-dimensional environments.
- **Advantage Actor Critic (A2C):** In the field of Reinforcement Learning, the Advantage Actor Critic (A2C) algorithm combines two of the RL algorithms which are Value-Based and Policy-Based algorithms. Value-Based algorithms learn their actions by using the predicted value of the state or action. Policy-Based agents learn a policy by mapping states to actions.
- **Trust Region Policy Optimization (TRPO):** TRPO updates policies by taking the most drastic step possible to increase performance while adhering to a strict limit on the distance between new and old policies. KL-Divergence, a measure of distance between probability distributions, is used to represent the restriction.

3 Methodology

In this section of the paper, the methodology followed during this research will be explained. This includes the use case and the MDP formulation.

3.1 Use case

Mediator determines the optimal actions when the driver becomes unfit because of fatigue. These actions will be explained in detail in the MDP formulation section.

Automation Levels

As mentioned above MEDIATOR project is focused on semi-automated vehicles. This means the system consists of several automation levels. These automation levels will be explained in the next section.

Automation levels in the MEDIATOR system can be summarised as follows:

- Manual Driving, the driver will handle all of the driving tasks and the automated system does not do anything related to driving. This automation level will be mentioned as L0 in the next parts of the paper.
- Partial Automation, the driver must remain engaged with the driving tasks and monitor the environment. This automation level will be mentioned as L2 in the next parts of the paper.
- Conditional Automation, the driver is not required to monitor the environment but must be ready to take over. This automation level will be mentioned as L3 in the next parts of the paper.
- High Automation, the vehicle can handle all driving situations and the driver can have the option to take over. This automation level will be mentioned as L4 in the next parts of the paper.

Note: Optimal automation mode will be mentioned as Lopt in the following sections. Optimal automation mode means the maximum automation level available at that moment.

Fatigue Levels

Karolinska Sleepiness Scale (KSS) will be used to determine fatigue levels. KSS is used to quantify the sleepiness level of the human driver. KSS levels will be assumed as follows:

- Alert (KSS = 1-4)
- Neither alert nor sleepy (KSS = 5)
- Signs of sleepiness (KSS = 6-7)
- Sleepy (KSS = 8-9)

In the concept of this research, some assumptions will be made. These assumptions will be explained in the next part.

Assumptions

- Only the case where fatigue occurs during manual mode will be taken care of. In this research, it is assumed that fatigue is a problem only when the fatigue occurs in manual mode.
- When the user fatigue level is "Alert" or "Neither Alert nor Sleepy" it is assumed that the driver can drive in manual mode, thus the mediator do nothing

- When the user fatigue level is "Signs of sleepiness" it is assumed that the driver will not be able to drive in manual thus mediator takes the action correct fatigue, suggest to shift to Lopt or shift to Lopt
- When the user fatigue level is "Sleepy" it is assumed that the driver cannot stay in manual driving and L2 and L3 are also not safe since the driver cannot be trusted neither to monitor the environment nor to take over thus the mediator takes the action shift to L4 or if this is not possible it takes the action emergency stop.

3.2 MDP Formulation

As it is explained above MDP formulation has 4 parts. Each of these parts and how they are defined in this research will be explained.

State Space

State Space consists 5 different parts. An overall representation of this is shown in the Figure 1. Each of these parts will be explained in detailed in the following part:

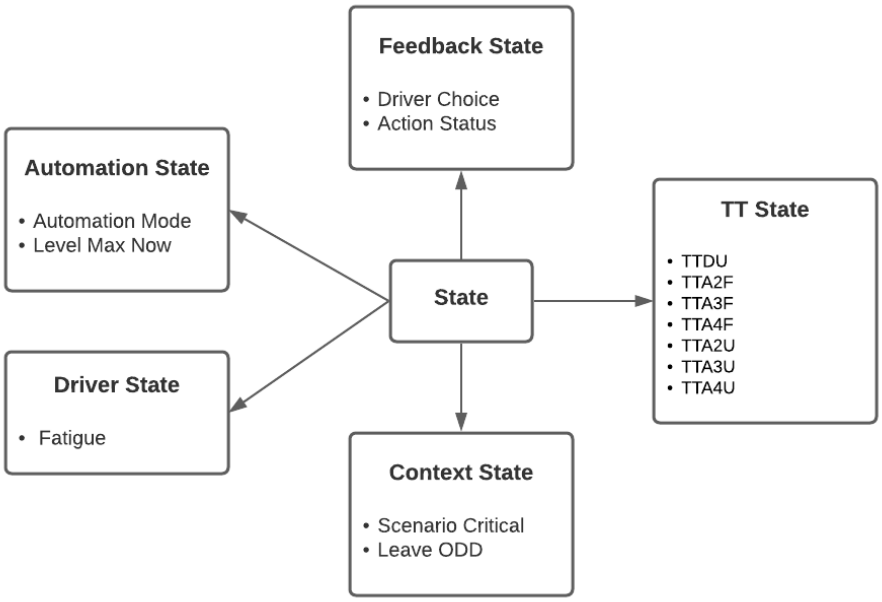

Figure 1: State Space

Driver State

There are two factors which can affect driver's state which are fatigue and distraction. But since distraction is not a relevant factor for this research's use-case this factor will be ignored. Thus, only fatigue will be used to represent driver state. Integer values will be used to represent fatigue levels. As explained above KSS will be used to determine fatigue

levels. "Neither alert nor sleepy" case will be ignored in this research since the action will be taken does not change in the cases of "Alert" and "Neither alert nor sleepy". In these cases, the action taken is always 'Do Nothing'. Thus fatigue levels will be as follows:

- 0: "Alert",
- 1: "Sign of Sleepiness",
- 2: "Sleepy"

Automation State

There are two factors affecting automation state and both of these factors are important for also use-case of this research. These two factors are: automation mode and the currently available maximum level of automation. Automation mode is the level of the automation currently used. These levels are explained above. If we recall them:

- L0 - Manual Driving
- L2 - Partial Automation
- L3 - Conditional Automation
- L4 - High Automation

TimeMetrics State

There are several timemetrics which can be used in timemetrics state. Metrics which are relevant for the use-case of this research are TTDU, TTAF, TTAU. Simple explanations of these factors is as follows:

- **TTDU:** This is the abbreviation for time to driver unfitness. Driver becomes unfit when degradation (e.g. fatigue) behavior occurs. The time to driver unfitness is defined as the estimated time until the driver is no longer able to safely perform the manual driving task and is most relevant while driving manually or in partial automation, where the driver is expected to continuously be involved in the driving task.
- **TTAF:** This is the abbreviation for time to automation fitness, which describes how much time is left before an automation level will become available. Automation fitness is mainly impacted by ODD.
- **TTAU:** This is the abbreviation for time to automation unfitness, which describes how much time is left before an automation level will no longer be available.

TTAF and TTAU will be split in three parts, namely TTA2F, TTA3F, TTA4F and TTA2U, TTA3U, TTA4U. The difference between these metrics are TTAxU or TTAxF represents the TTAU in the automation level Lx. For example, TTA2U is the TTAU when automation mode is L2.

Context State

Factors which are part of the context state are:

- **NDRT type:** non-driving related tasks (NDRTs) can affect the required takeover time in varying degrees, e.g., task modality influences take over time.
- **Uncomofort Events:** the uncomfortable events are likely to make driver feel uncomfortable
- **Sensor Failure:** the automated system meets a sensor failure or not
- **Leave ODD:** the vehicle is going to leave ODD in 5 mins

Only leave ODD will be considered in this usecase. Since leaving ODD will also affect the action which is going to be taken in the case of fatigue.

Feedback State

Feedback State consists of the following factors which are all relevant for the use-case of this research:

- **Driver Response:** decision logic generates a shift action with preference (moderate, strong, and enforced) to HMI, and HMI negotiates with the driver about the action. The feedbacks that HMI can provide to decision logic is the driver's choice. There are 3 options of the driver: "No Response", "Accept" or "Reject".
- **Last Action:** Last action taken by the mediator module.
- **Vehicle Stop:** This is a binary variable that can be 0 or 1. This value is updated and used to know whether an emergency stop (ES) is required. When the variable is 0 it means ES is not required and when it is 1 it indicates at an ES is required.

Action Space

Actions which are part of the actions space are designed to provide safety and comfort for the driver. There are 5 actions in the action space.

- **Do Nothing (DN):** this action means that the mediator does nothing to change the current state in which the driver is. Since safety and comfort conditions are already satisfied, the current situation is not wanted to be changed.
- **Correct Fatigue (CF):** this action means the mediator tries to correct fatigue. For this purpose, it tries to reduce the fatigue level of the driver. This can be accomplished in numerous ways such as warning the driver etc. But the decision logic part is not responsible for this and just decides to do this action. In this research, it is assumed that by 70% of chance this action will reduce the driver's fatigue level. So, if the mediator uses this action and reduces the fatigue level then the driver can continue in the manual mode.

- **Suggest Shift to Lopt (SSL):** Lopt has explained above as the optimal automation level. In this action, the mediator suggests shifting to Lopt, which is an automation mode such as (L2, L3, L4) since it is assumed that the driver cannot continue in manual mode. This is a suggestion thus after this action driver responds with his choice and this part has explained in the feedback state as driver choice.
- **Shift to Lopt (SL):** the difference between SSL and SL is in SSL the mediator suggests shifting and waits for the driver's response. But in SL it does not suggest, it is forced to shift to Lopt since the current situation is not safe and it is urgent to shift to provide comfort and safety for the driver.
- **Emergency Stop (ES):** this action is the action of stopping the car. In some cases, when no other option solves the critical situation, to keep the driver safe, the car needs to be stopped. And this is done by using this Emergency Stop action.

Transition Function

As explained in previous sections, the transition function is part of MDP which represents the probabilities of going from one state to another state by performing a specific action. Representation of transition function in the model of this research is represented in the Figure 2.

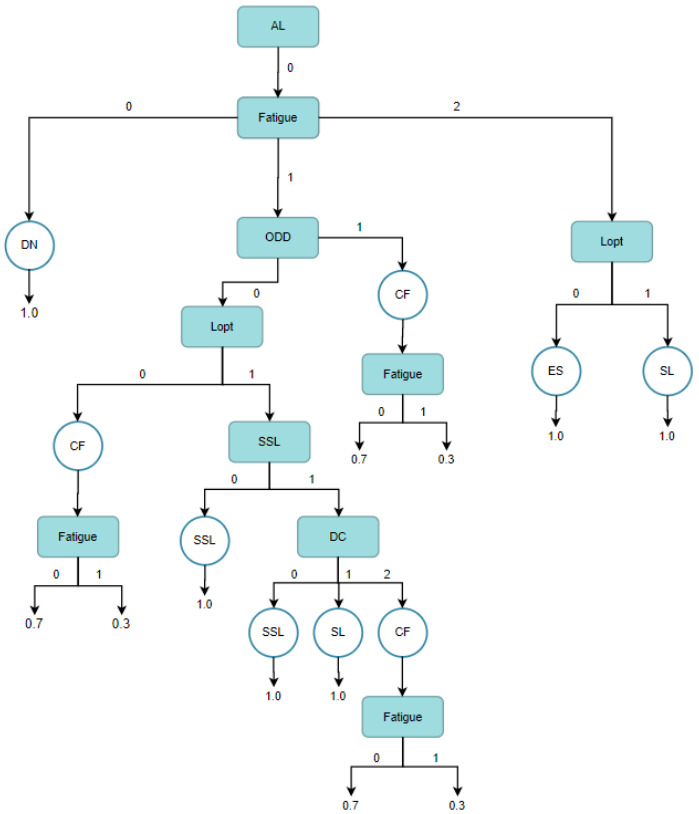

Figure 2: Transition Function

Figure 2 represents the transition function as a decision tree. The blue nodes represent the possible states. White nodes represent the actions that can be taken. The edges connecting them represent the possible values of the states. Finally, the values in the leaves of the tree represent the probability of this transition which is the probability of taking this action in this state. For example, in the right part of the tree, the AL level is 0 and the fatigue level is 2 if Lopt is 0 which means Lopt not available then doing ES action has the probability of 1.0. If the most left part of the tree is checked AL level is 0 and the fatigue level is 0 then DN action has the probability of 1.0. For an alternative version of the representation of this transition function, see Appendix A.

Reward Function

Reward function plays an important role in MDP formulation. This part affects how the model is going to learn which is the main purpose. To make it easier to understand, the reward function is represented as a decision tree in the Figure 3.

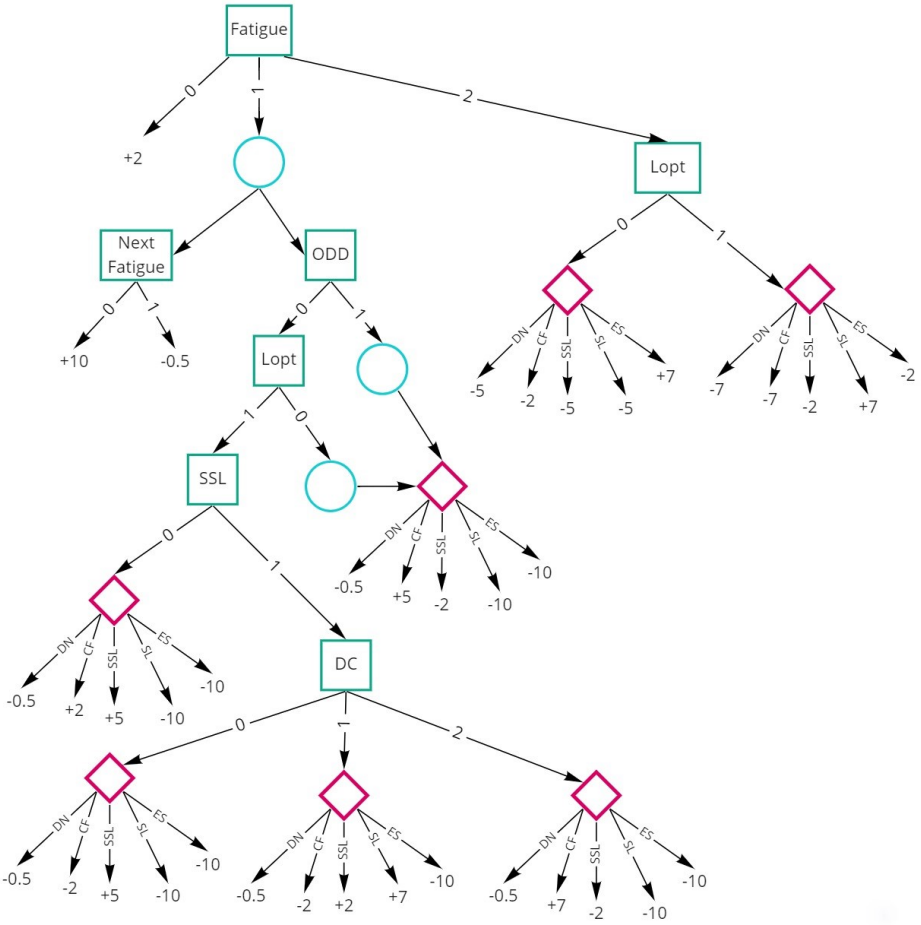

Figure 3: Decision Tree Representation of Reward Function

Figure 3 represents the reward function of the MDP formulation. Each node represents a state. Different nodes are connected with the edges representing possible situations the driver can be in. Pink nodes represent the different actions that can be taken by the mediator and finally each leave has the reward which represents the result of the designed reward function.

3.3 Baseline Policy

In this research, performance of the created MDP model will be compared with the performance of the decision tree baseline policy. Optimal decision tree baseline policy created for this use case is represented in Figure 4.

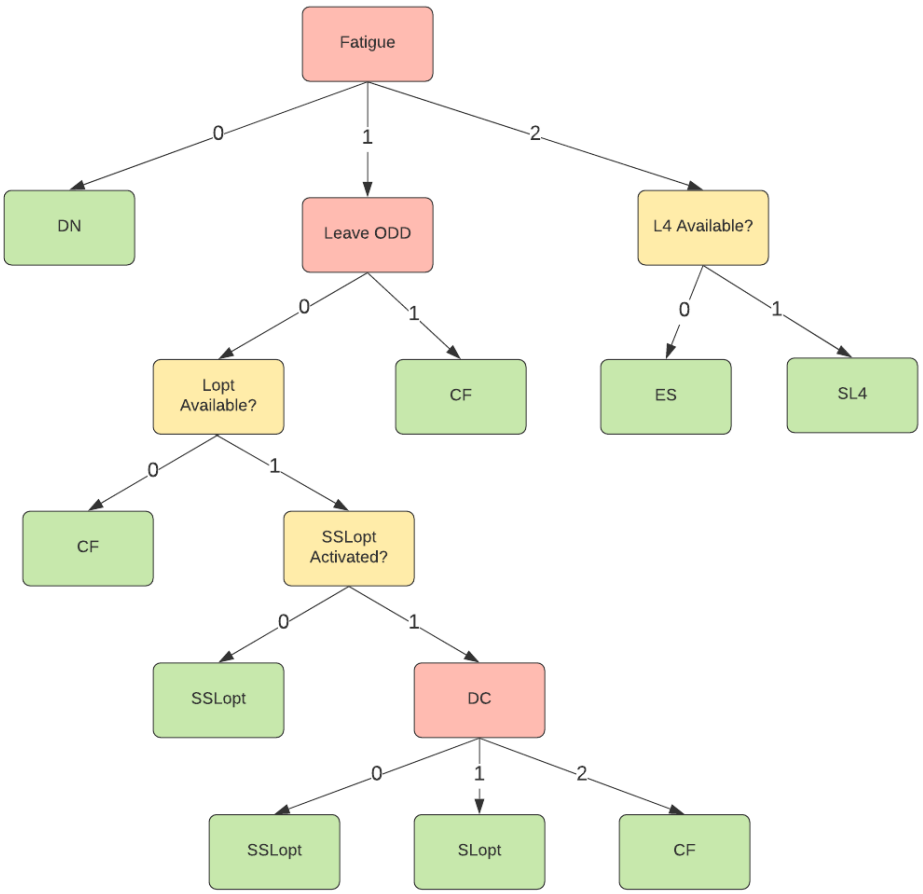

Figure 4: Baseline Policy

Figure 4 is a decision tree representation of the baseline policy where green nodes represent the actions that can be taken. Abbreviations of these actions are used since they are

already explained in the action space of the MDP formulation. Red nodes represent variables that can change in different states: fatigue, leave ODD and DC represents the driver choice. Yellow nodes represent whether that situation is available, for example, "Lopt available?" means whether the optimal level of automation is available or not. And "SSL activated?" means whether SSL action is called in the previous action.

Take a deeper look at what this tree represents: First of all, if the fatigue level is 0, the mediators does the DN action since there is no critical situation which need to fixed. If the fatigue level is 1, this means the driver is neither alert nor sleepy. Thus, first of all, it is checked whether the car will leave ODD. If it is, the mediator tries to correct the fatigue of the driver. If not leave ODD and Lopt is not available, since the mediator cannot shift to not available automation mode it still tries to correct fatigue. If Lopt available it is checked whether the SSL called in the last action, if not SSL action is done. Otherwise, the driver's choice is checked. If he gives no response (0) suggest action is done again, if he accepts (1) the mediator shifts to optimal automation mode. If he rejects (2) then as the last option, the mediator tries to correct the fatigue. The last case checked by this policy is when the fatigue level is 2. The mediator checks whether L4 is available since a sleepy driver cannot control other partial automation modes. If it is available it shifts to L4, otherwise, since the driver cannot continue driving it takes the ES action.

4 Results and Evaluation

The findings of this study will be discussed in this section. Following the formulation of the MDP model, which is detailed in the preceding section, various existing RL algorithms will be applied. The next section will go over these RL algorithms. Following this section, the choice of evaluation metrics will be discussed, followed by the model's outcomes based on these metrics.

4.1 Experimental Setup

To evaluate the performance of the created MDP model, different RL algorithms are used. Namely: DQN, A2C, and TRPO. These algorithms are chosen since all of these algorithms are supported by OpenAI Baselines which can be used easily to test the MDP model. Each of these algorithms is run 1 million timesteps. While running the algorithms different hyperparameters are used to make their performance better. These hyperparamters are as follows:

- **DQN:** total timesteps is 1000000 , buffer size is 5000, exploration fraction is 0.1, exploration final episode is 0.02, training frequency is 25, batch size is 128, print frequency is 100, learning start is 1000, gamma 0.1, and target network update frequency is equal to 100.
- **A2C:** total timesteps is 1000000, and gamma is equal to 0.1.
- **TRPO:** total timesteps is 1000000, learning rate is 0.001, number of steps is 50, schedule of learning rate is 'linear', and gamma is equal to 0.1.

4.2 Evaluation Metrics

Since the mediator needs to ensure driver safety and comfort. Several evaluation metrics are determined. These metrics are split into 3 parts: driving safety, driving comfort, and decision efficiency:

Driving Safety

- **Number of unsafe actions:** The number of unsafe actions includes the following:
  - Manual driving with TTDU=0, i.e., driver unfit
  - L2/L3/L4 on with TTAU=0, i.e., automation unfit
  - Shift to automation when TTAF!=0
- **Fixed scenarios:** Fixed scenarios are the scenarios where there was a critical scenario and it no more exists. Fixed Scenarios/Fatigue Critical Scenarios will be calculated. This gives the success rate.
- **Duration of being in the highest fatigue level:** This metric represents the duration when the driver is in the highest fatigue level (very sleepy). This metrics will be represented as DHFL.

Driving Comfort

- **Duration of being in fatigue:** The duration of being in fatigue critical situation. In this case, the longer the duration is, the lower the comfort is.

Decision Efficiency

- **Time to Solve Critical Scenario:** This is the time to solve the scenario in the fatigue critical situation. Solving scenario means waking up the driver or shifting to Lopt.
- **Action distribution:** Action distribution is the distribution of actions according to fatigue levels. In each fatigue level, desired actions are different.

4.3 Results

In this part, the results of the RL algorithms will be explained considering different aspects metrics. Note that best performances are shown as bold and worse ones as italic in the tables.

Driving Safety

Unsafe actions are represented as the percentage and duration in the highest fatigue level is as seconds since it is easier to grasp it in this way. Lastly, fixed scenarios is the percentage of unsafe scenarios which are fixed. This is represented as the percentage.

Algorithms	Unsafe Actions (%)	DHFL (s)	Fixed Scenarios (%)
baseline	3.80	1.34	36.53
DQN	3.22	**1.29**	**87.53**
A2C	*23.06*	*2.58*	*16.97*
TRPO	**3.21**	1.30	67.16

Table 1: Driving Safety Metrics

Table 1 shows that overall DQN performs best in terms of driving safety. Since it has a high percentage of fixed scenarios. They perform closely with the TRPO algorithm in case of unsafe actions percentage and duration in the highest fatigue level. Baseline policy also has closer results with the other 2 best-performing algorithms in case of unsafe actions and duration in the highest fatigue level. But it performs poorly on the fixed scenarios metric. Finally, A2C is the worse performing algorithm here in terms of driving safety.

Driving Comfort

Duration of being in fatigue will be represented as percentage. This is ratio of states with fatigue to the whole states.

Algorithms	Duration of Being in Fatigue (%)
baseline	2.74
DQN	1.58
A2C	*5.89*
TRPO	**1.49**

Table 2: Driving Comfort Metrics

From Table 2, it can be concluded that TRPO is the best algorithm in terms of driving comfort but its performance is close to DQN thus, it can be said that DQN also provides comfort for the driver. Both of these algorithms perform better than baselines in the case of driving comfort. But A2C performs poorly according to results.

Decision Efficiency

Time to solve critical scenario is represented in seconds. This has been calculated by measuring the time until the mediator wakes up the driver or shifts to Lopt.

Algorithms	Time to Solve Critical Scenario (s)
baseline	*2.80*
DQN	2.67
A2C	**1.63**
TRPO	2.64

Table 3: Decision Efficiency Metrics

Table 3 shows that A2C is good in terms of the decision efficiency, which means it takes less time for A2C to decide to the necessary action. Other three metrics performs closely

with each other.

Action distribution is the percentage of different actions in different fatigue levels. This is shown with histograms in Figure 5 to visualize it better. According to results obtained in Figure 5, it can be said that algorithms except A2C acts as expected. A2C does not work as expected since it performs CF action so often when fatigue is 0, which does not make sense. Since, the driver is alert. Also, it mostly performs DN action in case of fatigue 1 when user is neither alert nor sleepy. In this situation expected action is to correct fatigue or shift to an automation mode. Except for A2C, the action distributions of the other methods are similar. One difference that takes attention is when the fatigue level is 2, DQN and TRPO mostly make ES action but the baselines method also tries to do the SL action.

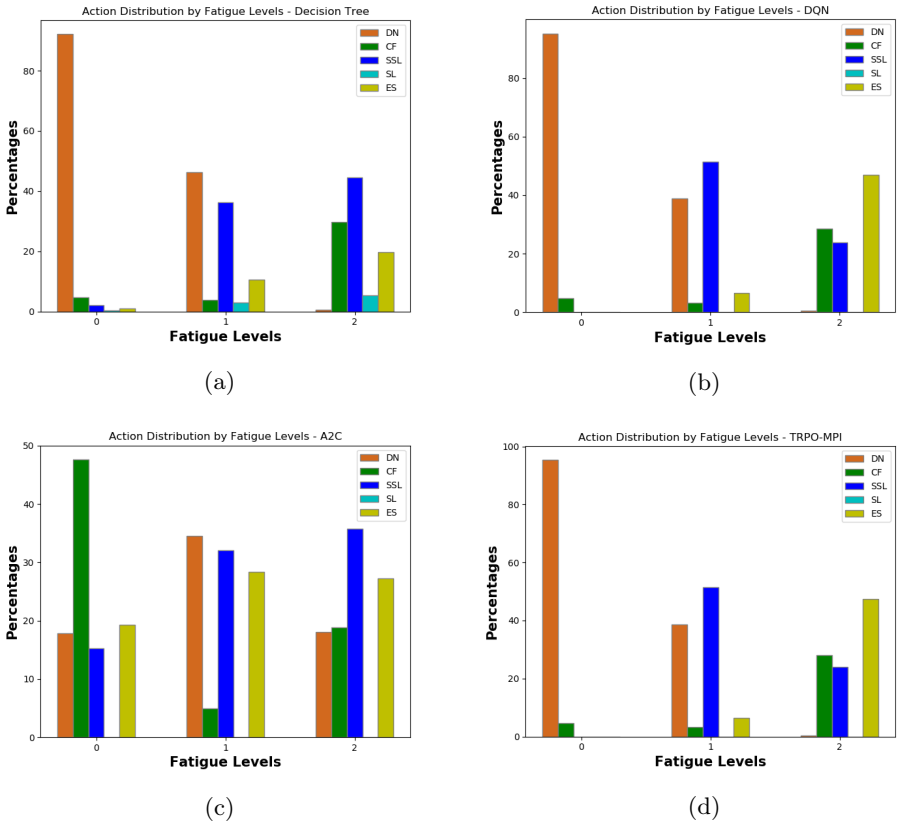

Figure 5: Action Distribution Histograms

5 Discussion

Following a thorough examination of the previous section's findings, it is clear that DQN and TRPO outperform the baseline policy in terms of driving safety, driving comfort, and decision efficiency, as shown in tables 1 - 3. A2C, in contrast to others, does not perform well in terms of driving safety and comfort. When analyzing the results of action distribution,

it was concluded that A2C does not perform well in terms of decision efficiency. This is clearly visible in Figure 5. It has been concluded that DQN and TRPO outperform other algorithms. If the two are compared, DQN arguably could be the better and safer algorithm since it has a higher percentage of fixed scenarios as shown in Table 1. These results could be improved more by running much more times and by analyzing in which situation DQN performs better and in which situation TRPO is more successful.

6 Responsible Research

Responsible research is an important part of conducting research. This section will be focused on the ethical concerns of this research and will be explained in 2 parts, namely: ethical aspects and reproducibility.

6.1 Ethical Aspects

During the experiment process, no real-life experiments are conducted, thus no real-life data collected. As a result, ethical and privacy concerns do not apply to this research. Still, there are some assumptions made during this research. This is mainly the case of deciding to the correct actions in different states. But from an ethical aspect, it cannot always be assumes that desired actions of different drivers are the same. For example, in the case of a high fatigue level (sleepy), the mediator takes the ES action and stops the car. But this would be a not desired case for some drivers.

The following is another ethical aspect of the research: As previously stated, the goal of this research is to provide driver safety and comfort. And, if this model would be used in real life in the future, the mediators must ensure that these safety and comfort constraints are met. However, the existing model can not provide a 100% safety guarantee to the driver. Thus the drivers should be informed about this. Any crashes or issues that arise as a result of poor decisions have a major ethical implication for driver's safety that must be addressed.

6.2 Reproducibility

For the reproducibility aspect of this research, The Machine Learning Reproducibility Checklist by Joelle Pineau is used [14] to make sure that this research is reproducible. For the models produced in this research such as MDP formulation, a clear description of the algorithm/model is included. All of the assumptions made during research are explained. About the theoretical claims made, all of them are stated and proved accordingly. No database is used in this research thus, that is not an applicable case to check the reproducibility of this research. The input data for this research is simulated rather than using a database. In the experimental results, the range of hyper-parameters, the exact number of training and evaluation runs, definitions of the measures used to gather statistics, and finally description and evaluation of results are shared. Lastly, the source code of the project can be shared on request. Also all of the algorithms used are open source libraries which also contributes to the reproducibility. All of these facts indicate that this research is reproducible.

7 Conclusion

In conclusion, the goal of this research is to model the fatigue use case of semi-automated driving as an MDP to allow the mediator to do the correct action and provide safety and comfort to the driver. This model is evaluated by using existing RL algorithms and compared with the optimal baseline method. Some of these algorithms (DQN and TRPO) performed better than the baselines policy. Thus, it has been shown that a successful model can be created for this use case. Although this model performs well by evaluating with these algorithms, this provided model still does not fully ensure driver safety and comfort. To improve performance more other unexplored existing RL algorithms can be tried and also the model can be improved further.

References

[1] Mediator. (2021). MEDIATOR. Retrieved May 26, 2021, from https://mediatorproject.eu/about/about-mediator.

[2] van Wyk, F., Khojandi, A., & Masoud, N. (2020). Optimal switching policy between driving entities in semi-autonomous vehicles. Transportation Research Part C: Emerging Technologies, 114, 517-531.

[3] Vermunt, D. (2020). A Markov Decision Process approach to human-autonomous driving control logic. Delft University of Technology, 0-78.

[4] Ahlström, C., Zemblys, R., Jansson, H., Forsberg, C., Karlsson, J., & Anund, A. (2021). Effects of partially automated driving on the development of driver sleepiness. Accident Analysis & Prevention, 153, 106058.

[5] Jarosch, O., Bellem, H., & Bengler, K. (2019). Effects of Task-Induced Fatigue in Prolonged Conditional Automated Driving. Human Factors, 61(7), 1186-1199.

[6] Wang, Y., & Ma, J. (2018). Automation Detection of Driver Fatigue Using Visual Behavior Variables. Archives of Civil Engineering, 64(2), 175-185.

[7] Desmond, P. A., Hancock, P. A., & Monette, J. L. (1998). Fatigue and Automation-Induced Impairments in Simulated Driving Performance. Transportation Research Record, 1628(1), 8-14.

[8] Vogelpohl, T., Kühn, M., Hummel, T., & Vollrath, M. (2019). Asleep at the automated wheel-Sleepiness and fatigue during highly automated driving. Accident; analysis and prevention, 126, 70-84.

[9] Dawson, D., Reynolds, A. C., Van Dongen, H. P. A., & Thomas, M. J. W. (2018). Determining the likelihood that fatigue was present in a road accident: A theoretical review and suggested accident taxonomy. Sleep Medicine Reviews, 42, 202-210.

[10] Foy, P. (2021). Deep Reinforcement Learning: Guide to Deep Q-Learning. MLQ. Retrieved May 27, 2021, from https://www.mlq.ai/deep-reinforcement-learning-q-learning

[11] Wang, M. (2021). Advantage Actor Critic Tutorial: minA2C - Towards Data Science. Medium. Retrieved May 27, 2021, from https://towardsdatascience.com/advantage-actor-critic-tutorial-mina2c-7a3249962fc8

[12] Sidor, S. (2020). OpenAI Baselines: DQN. OpenAI. Retrieved May 27, 2021, from https://openai.com/blog/openai-baselines-dqn

[13] Reinforcement Learning (DQN) Tutorial. (2021). PyTorch. Retrieved May 27, 2021, from https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html

[14] Pineau, J. (2020). The Machine Learning Reproducibility Checklist. McGill University. https://www.cs.mcgill.ca/ jpineau/ReproducibilityChecklist.pdf

[15] Trust Region Policy Optimization - Spinning Up documentation. (2018). OpenAI Spinning Up. Retrieved June 7, 2021, from https://spinningup.openai.com/en/latest/algorithms/trpo.html

A Appendix

State						Action	Result State						Probability
F	AL	ODD	Lopt	SSL	DC		F	AL	ODD	Lopt	SSL	DC	
0	0					DN	0	0					1.0
1	0	0	0			CF	0	0	0	0			0.7
1	0	0	0			CF	1	0	0	0			0.3
1	0	0	1	0		SSL	1	0	0	1	0		1.0
1	0	0	1	1	0	SSL	1	0	0	1	1	0	1.0
1	0	0	1	1	1	SL	1	Lopt	0	1	1	1	1.0
1	0	0	1	1	2	CF	0	0	0	1	1	2	0.7
1	0	0	1	1	2	CF	1	0	0	1	1	2	0.3
1	0	1				CF	0	0	1				0.7
1	0	1				CF	1	0	1				0.3
2	0		0			ES	1						1.0
2	0		1			SL	2	4		1			1.0

Table 4: Transition Function of MDP