Lane-Change Decision-Making for Autonomous Vehicles

A Comparative Study of Rule-Based and Machine Learning Models.

Faton Hoti, Balaji Sathiya Venkata Narayanan, Pavan Kumar Adiga Nagaraj, Sanjana Hassan Ananda Kumar, Mattia Carlino, Alexander Ybring, Elias Kihlström

Abstract

Autonomous driving systems are at the forefront of modern transportation, aiming to enhance mobility, reduce accidents, and minimize human errors. A critical function of these systems is lane-change decision-making (LCDM), which involves navigating complex and dynamic traffic scenarios while ensuring safety and efficiency. Traditional rule-based LCDM frameworks are valued for their interpretability but often struggle with scalability and adaptability to real-world conditions. Machine learning (ML) models, on the other hand, excel in handling complexity and unpredictability but suffer from a lack of interpretability, posing challenges for regulatory approval and trustworthiness. This project presents a comparative analysis between different variants of the two LCDM models Liu et al. and SL2015, exploring the trade-offs between interpretability, adaptability, and performance in different traffic scenarios. Leveraging the Simulation of Urban Mobility (SUMO) tool, we evaluate pre-existing rule-based and ML-enhanced frameworks, such as the hybrid approach proposed by Liu et al. [1], under diverse traffic conditions. The results demonstrate that while ML models provide higher adaptability and performance in complex scenarios, rule-based models offer better transparency and predictability. The hybrid approach achieves a balance, providing both improved performance and moderate interpretability. While these findings provide valuable insights, further work is needed to refine and validate the results. Nevertheless, the study offers a clearer direction for identifying effective and ineffective approaches.

Keywords: Lane-Change Decision-Making (LCDM), Rule-Based Frameworks, Interpretability, Hybrid Decision-Making-Models.

Table of Contents

1 Introduction

Along with the artificial intelligence (AI) boom the field of autonomous vehicular systems has also seen rapid growth with companies such as Tesla leading the charge. Autonomous driving represents modern transportation and aims to enhance mobility, reduce accidents, minimize human mistakes, and much more. A core function of autonomous vehicles is their ability to apply critical thinking in the form of lane-change decision making (LCDM). Making safe and efficient lange changes is crucial for vehicles to be able to safely navigate from A to B. In practice autonomous vehicles require integration of many interconnected systems such as cameras, LiDAR, and radar to capture information about the near environment which can be used in its decision making. Lane-changing is an inherently complex concept as it requires taking into account many factors in relation to each other. In dynamic and unpredictable traffic environments it is not trivial to take into account factors such as surrounding traffic behaviours, pedestrians, vehicle dynamics, roadway conditions and more. LCDM has been extensively researched within the context of autonmomous vehicle control systems. Before the emergence of AI as a mainstream tool, rule-based frameworks set the foundation. They have several advantages such as being deterministic and interpretable, which is important because it allows us humans to understand the process rather than it being a black box. An inherent weakness of rule-based frameworks is that they tend to struggle with complexity as well as generalize poorly, meaning they do not perform equally well in most scenarios. It is for such reasons that rule-based frameworks are often combined with machine learning (ML) algorithms. A recent example is highlighted by e.g. Liu et al. [1] where they combined a rule-based framework in combination with the supervised learning algorithm support vector machines (SVMs) in their LCDM. Such models show an interesting potential in managing complex traffic scenarios with respect to complexity and unpredictability by optimizing the decision-making process with the help of machine learning tools. However, the introduction of ML tools also has a drawback. Due to the *black box* nature of such tools, the interpretability of the resulting models suffers, which is one of the key difficulties of developing LCDM models which limits their acceptance in safety-critical applications. Note that the requirement of interpretability is not just a technical requirement, but also a requirement for regulatory approval of the autonomous system at hand.

This project provides a comparative analysis of a few hand-picked LCDM models. The primary focus is to address the trade-offs between interpretability, adaptability, and performance in different traffic scenarios. By defining realistic lane-changing scenarios, implementing and evaluating multiple LCDM models, and simulating them using tools such as Simulation of Urban Mobility (SUMO), the comparative study seeks to identify models that strike a balance between efficiency, safety, and interpretability. Due to the inherent difficulty of making up own LCDM models in combination with the lack of domain knowledge within the group, the study builds on existing work, such as that of Liu et al. [1], exploring enhancements to preexisting models and evaluating their performance against each other. This research contributes to the advancement of intelligent transportation systems by addressing a key challenge in autonomous vehicle

operation: reliable and interpretable lane-changing decision making.

2 Methodology

This study aims to compare the performance of lane-changing decision-making models based on machine learning and rule-based approaches. The evaluation is carried out on three highway scenarios to analyze the behavior of the model under varying traffic conditions. The primary focus of these scenarios is the lane change of an ego vehicle from the fastest lane to the slowest lane on a 4-lane highway, followed by the vehicle exiting the highway.

Scenario A represents a low-traffic-density condition, Scenario B simulates moderate traffic, and Scenario C involves high traffic density. These scenarios are meticulously developed, designed using coding, and implemented using the Simulation of Urban Mobility (SUMO) tool, leveraging its capabilities to model realistic traffic dynamics and conditions.

In this study the TraCI (Traffic Control Interface) is used as a crucial component for integrating external control logic with the SUMO simulation environment. TraCI enables a real-time interconnection with the simulation, providing fine control over simulation elements mainly the Ego vehicle, traffic flow, and the network properties. TraCI is employed to implement and evaluate the lane-changing decision-making models. Specifically:

- Model Execution: TraCI facilitates the integration of external machine learning models or rule-based algorithms for lane-changing decisions, allowing these models to influence the behavior of the ego vehicle dynamically during the simulation.
- Scenario Control: It enables the dynamic adjustment of traffic conditions, such as modifying vehicle speeds, positions, and densities, to ensure accurate representation of the low, moderate, and high traffic scenarios.
- Data Extraction: TraCI provides real-time access to simulation data, such as vehicle states, point of no return for steering and braking, lateral acceleration, fuel consumption, the relative speed of the ego vehicle and its neighboring vehicles, and traffic flow before the lane change and after the lane change, which is essential for evaluating and comparing the performance of the models.
- Custom Behavior: TraCI allows customization of the ego vehicle's lane-change logic by overriding SUMO's default behavior, enabling precise implementation of both machine learning and rule-based models.

2.1 Models

This subsection presents an overview of the models examined in this study, encompassing both re-implementations of established methods and original contributions. The

SL2015 model is based on a re-implementation of SUMO's sublane framework. The Liu et al. model is analyzed in its original form and is further enhanced with adaptive coefficients and dynamic safety adjustments, reflecting our original improvements to address limitations in adaptability and robustness. Lastly, a Support Vector Machine (SVM) model inspired by Liu et al.'s methodology is implemented and adapted for evaluation. Together, these models highlight the interplay of rule-based, hybrid, and machine learning approaches to lane-change decision-making.

SL2015 model

The following custom implementation reimagines the SL2015 sublane model from SUMO (Simulation of Urban MObility), hence providing a complete framework for lane change decision-making that evaluates a sequence of weighted criteria for optimal lane positioning.
The model integrates three behavioral parameters with equal weighting:

$$STRATEGIC\_PARAM = 1.0, \quad \text{representing strategic positioning;}$$

$$SPEEDGAIN\_PARAM = 1.0, \quad \text{representing speed optimization;}$$

$$KEEPRIGHT\_PARAM = 1.0, \quad \text{representing lane preference maintenance.}$$

These parameters operate in conjunction with strict safety constraints to ensure reliable lane-changing decisions.
The safety verification of the model imposes specific spatial and temporal constraints through two key parameters: the enforced minimum safe gap,

$$MIN\_SAFE\_GAP = 15\,\text{meters},$$

and the defined reaction time,

$$REACTION\_TIME = 2.0\,\text{seconds}.$$

The safety evaluation for forward and rear gaps is conducted separately, with each check performed conditionally based on the presence of a vehicle in the desired lane. If no leading vehicle is present in the designated lane, the forward gap assessment is omitted. Conversely, if no following vehicle is present, only the rear gap assessment is excluded.
With respect to forward gaps, given a leading vehicle, the model directly implements the minimum safe gap requirement:

$$gap\_front \geq MIN\_SAFE\_GAP.$$

For rear gaps, when another vehicle is close, the safety check involves two conditions:
1. The static minimum gap requirement:

$$gap\_rear \geq MIN\_SAFE\_GAP.$$

2. A dynamic gap condition, which applies when the approaching vehicle is traveling at a high speed:

$$gap\_rear \geq REACTION\_TIME \cdot (follower\_speed - ego\_speed).$$

Any violation of the corresponding safety conditions prevents lane changes from being executed [2].
The behavior of the model is carefully controlled by a wide range of parameters related to lane changes [3]. The properties of lateral motion are determined by:

$$lcSublane = 0.5, \quad \text{the sublane position;}$$
$$lcAssertive = 0.2, \quad \text{the assertiveness level;}$$
$$lcAccelLat = 0.5, \quad \text{the lateral acceleration limit.}$$

In addition, the model includes a dynamic impatience factor:

$$lcImpatience = 0.0,$$

which builds up over a specified time:

$$lcTimeToImpatience = 120.0\,\text{s}.$$

Supplementary parameters are added to the model to control various aspects of lateral behavior:

$$maxSpeedLat = 0.5, \quad \text{the maximum lateral velocity;}$$
$$minGapLat = 1.5, \quad \text{the minimum lateral spacing;}$$
$$latAlignment = \text{“center”}, \quad \text{the preference for lateral alignment.}$$

Additionally, a parameter controlling gap acceptance, called pushiness, is introduced [3]:

$$lcPushy = 0.1.$$

Strategic lane change decision-making component: Uses a ternary scoring system based on lane indices. If the current and desired lanes are the same, it returns 0.0 when evaluating a potential lane change. It assigns a score of 1.0 if the target lane index is lower than the current lane index and -1.0 if it is higher. This can be expressed formally as:

$$\textbf{score} = \begin{cases} 0.0 & \text{if } current\_lane = target\_lane, \\ 1.0 & \text{if } target\_lane < current\_lane, \\ -1.0 & \text{if } target\_lane > current\_lane. \end{cases}$$

The model learns to assign the lane indices by parsing the lane names, e.g., splitting by _ to extract numerical indices. This realizes a simple yet efficient strategy to maintain an optimal lane position with respect to the vehicle's goals.

It also includes the speed gain calculation considering the achievable speeds in the current and target lanes. The achievable speed for each lane is, by default, set to the maximum speed of the ego vehicle if there is no leading vehicle. If there is a leading vehicle, then the achievable speed is calculated as

$$v_{\text{achievable}} = \min(v_{\text{max}}^{\text{ego}}, v_{\text{lead}})$$

where v_{lead} is the actual speed of the leading vehicle. The speed gain incentive is then calculated as

$$\Delta v = v_{\text{achievable}}^{\text{target}} - v_{\text{achievable}}^{\text{current}}$$

which provides a direct measure of the potential velocity advantage of changing lanes. The keep-right incentive uses a similar binary decision-making process as the strategic component. The model implements the following decision rule:

$$\text{keep\_right\_score} = \begin{cases} 1.0 & \text{if } \textit{target\_lane} < \textit{current\_lane} \\ -1.0 & \text{otherwise} \end{cases}$$

This formulation creates a clear bias in maintaining rightward alignment whenever other factors would allow it.
The comprehensive incentive calculation combines these components into a weighted sum that determines the final lane-change decision:

$$\text{total\_incentive} = \alpha_s \cdot \text{strategic} + \alpha_v \cdot \text{speed\_gain} + \alpha_r \cdot \text{keep\_right}$$

$$\text{where:} \quad \begin{cases} \alpha_s & \text{represents the STRATEGIC\_PARAM} \\ \alpha_v & \text{represents the SPEEDGAIN\_PARAM} \\ \alpha_r & \text{represents the KEEPRIGHT\_PARAM} \end{cases}$$

A lane change is executed only when the safety conditions are met and the total incentive is greater than zero, thereby guaranteeing that lane changes are safe and beneficial to the objectives of the vehicle [2].

Liu et.al. and our enhancement

The original model [4] evaluates lane-change decisions based on benefit, tolerance, and safety functions. It is quite rudimentary and does not allow for dynamic changes, as well as does not generalize well to arbitrary scenarios. It is in nature a simple rule-based model that uses factors such as gaps between vehicles and speeds in combination with threshold values to make a decision. To increase adaptability and robustness in dynamic traffic environments and different scenarios, we introduced several key modifications to the Liu et al. model. First, we implemented adaptive coefficients that dynamically adjust based on real-time traffic density (ρ) and the speed of the ego vehicle (v_E). These coefficients are defined as follows:

- A: This coefficient scales the influence of the benefit function, quantifying the potential gain of the ego vehicle from changing lane. A increases as traffic density in the target lane increases.

$$A = 1 + 0.1 \cdot \rho$$

- B: This coefficient adjusts the impact of the speed difference between the desired speed (v_{set}) and the current speed (v_E). It ensures that the benefit function considers the gap between the current speed of the ego vehicle and the desired speed.

$$B = 1 + 0.05 \cdot (v_{\text{set}} - v_E)$$

- C: This constant coefficient influences the tolerance function, ensuring that the gap (G_p) between the ego vehicle and the trailing vehicle in the current lane is appropriately weighted. The tolerance function is defined as:

$$F_{\text{tol}} = C \cdot (G_p - v_E \cdot t_h)$$

  where t_h is the time progress.

- D: This coefficient modifies the safety function, increasing its value as the density of traffic increases. A higher D in dense traffic prioritizes maintaining safe gaps in the target lane.

$$D = 1 + 0.1 \cdot \rho$$

- E: This constant coefficient balances the relative speeds between the ego vehicle and the vehicles in the target lane in the safety function.

Traffic density (ρ) is defined as the number of vehicles in the target lane divided by the length of the lane:

$$\rho_{\text{target lane}} = \frac{N_{\text{target lane}}}{L_{\text{target lane}}}$$

Second, the benefit function (F_{ben}) was refined to consider the relative gap between the ego vehicle and vehicles in both the current and target lanes:

$$F_{\text{ben}} = A \cdot (v_{\text{set}} - v_p) + B \cdot (G_{\text{tp}} - G_p)$$

where v_p is the speed of the leading vehicle in the current lane, G_{tp} is the gap to the leading vehicle in the target lane, and G_p is the gap to the trailing vehicle in the current lane. This modification improves the model's ability to assess the availability of safe gaps for lane changes. Third, the safety function (F_{safety}) was improved to include a dynamic safe gap for trailing vehicles in the target lane introducing a dynamic safe gap:

$$G_{\text{tr,min}} = G_{\text{tr,default}} + 2 \cdot |v_E - v_{\text{TR}}|$$

where ($G_{\text{tr,default}}$) is the default minimum safe gap and v_{TR} is the speed of the trailing vehicle in the target lane. The safety function is then calculated as:

$$F_{\text{safety}} = D \cdot \max(G_{\text{tr}} - G_{\text{tr,min}}, 0) + E \cdot (v_E - v_{\text{TR}})$$

By dynamically adjusting the minimum safe gap based on the relative speed difference, the model prevents unsafe lane changes when vehicles in the target lane approach too quickly from behind. Finally, we introduced a mechanism to balance acceleration and deceleration strategies. If the ego vehicle cannot safely accelerate to overtake vehicles in the target lane, the model considers controlled deceleration to create a safe gap for a lane change. This adaptive speed strategy addresses scenarios where vehicles might otherwise remain stuck due to insufficient gaps. These modifications aim to improve the efficiency and reliability of the decision-making process, addressing limitations in the original Liu et al. model. Their actual impact on performance will be evaluated in the results section.

Support Vector Machine (SVM) Model

In the underlying paper (Lui et al [4]) they propose a multitude of SVM models utilizing different kernels to reach different accuracies. In this project, an SVM model heavily influenced by the SVM model from the paper has been implemented. SVM utilizes hyperplanes to separate data points that can be divided in multiple dimensions. The ultimate goal is to create a plane with the maximum distance to the surrounding data points. When the data is non-linear a kernel is used to compute the data in a higher dimension and the choice of the kernel can be fitted to the specific task [5]. Taking a similar approach as Lui et al [4] as well as [6], the NGSIM dataset was chosen and since the scenarios are based on highways the US 101 Highway and I80 Freeway datasets were selected. First, data points were extracted according to [4] which identifies the start of a lane change when the lane ID changes and the velocity increases by 0.6096 m/s. The point right before the beginning of a lane change was also selected to be when the lane ID changed. The preprocessing also consisted of calculating the distances of importance in the baseline model (P, G_{TR}, G_{TP}) which correspond to the preceding vehicle in the same lane, the following vehicle in the adjacent lane and the preceding vehicle in the adjacent lane respectively. Their velocities are also calculated and correspond to (v_P, v_{TR}, v_{TP}) the accuracy and velocity of the ego vehicle is a_E and v_E. However, to balance out the dataset the lane changes when there was no car in the adjacent lane were removed.

The data was then split into a training set and a testing set with the sizes 70% and 30% of the original dataset respectively. Using the sklearn [7] Standard Scaler, the extracted data was normalized so that the mean was close to 0 and the standard deviation was close to 1. Different C values were tested for different kernels and the Radial Basis Function (RBF) kernel was the best fit for the model. The accuracy on the test set for different C values can be seen in Figure 1 and 1 where $C = 0.04$ performed the best providing the model with a test set accuracy of approximately 79.07% which is a little less to the best accuracy achieved in [4] which reached a max accuracy of 86.27% using an optimized radial basis function as kernel. The confusion matrix for the test set can be seen in Figure 3. In Figure 2 the Recall, Precision, and F1-Score are also plotted. For the specific C value we have a much larger Recall value compared to the Precision value.

When running the model in SUMO the values for distances to the surrounding cars and their respective velocities as well as the acceleration and velocity of the ego vehicle are

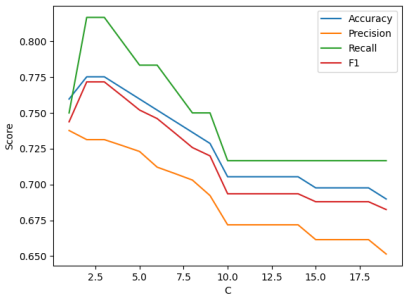

Figure 1: Recall, Precision, F1-Score, and Accuracy computed for different values of C. Where C scales from 1 to 10.

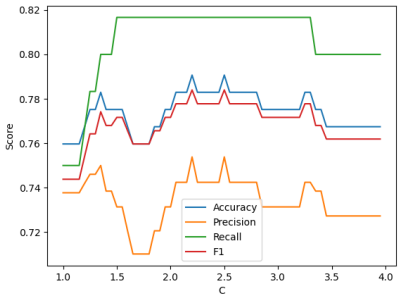

Figure 2: Recall, Precision, F1-Score, and Accuracy computed for different values of C. Where C scales from 0.01 to 1.0.

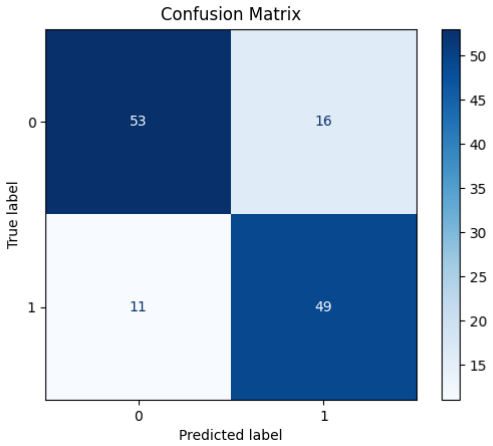

Figure 3: The confusion matrix for the test set in SVM using an RBF kernel and $C = 0.04$. ”0” means no to not make a lane change while ”1” is the decision to make a lane change.

extracted and used in the imported SVM model which then triggers a lane change or not.

2.2 Scenarios

The models are evaluated under three different scenarios, with traffic density being the primary variable. To ensure robustness and effectiveness in decision-making, the traffic includes not only cars but also buses and trucks.
Each scenario is structured around a 4000-meter highway with four lanes. Lane 1 has a speed limit of 50 km/h, Lane 2 has a speed limit of 80 km/h, Lane 3 has a speed limit of 100 km/h, and Lane 4 has a speed limit of 120 km/h. The highway exit is located 2000 meters from the starting point. The exit lane consists of two lanes, and no speed limit is defined for this exit road.
Scenario A :
Description: Ego vehicle to move from the fastest lane (lane 4) to the slowest lane (lane 1) and exit on the right-hand side exitway, within the defined simulation time.
Environment: Highway with an exit on the right-hand side with low traffic density. Traffic flow is estimated to be around 432 vehicles per hour. Vehicle types include car, bus, and truck.

Scenario B:
Description: Ego vehicle to move from the fastest lane (lane 4) to the slowest lane (lane 1) and exit on the right-hand side exitway, within the defined simulation time.
Environment: A Highway with a Medium traffic density exit on the right-hand side. Traffic flow is estimated to be around 504 vehicles per hour. Vehicle types include car, bus, and truck.

Scenario C:
Description: Ego vehicle to move from the fastest lane (lane 4) to the slowest lane (lane 1) and exit on the right-hand side exitway, within the defined simulation time.
Environment: A Highway with a High traffic density exit on the right-hand side. Traffic flow is estimated to be around 2000 vehicles per hour. Vehicle types include car, bus, Motorcycle and truck.

2.3 Evaluation Criteria

The performance of the lane-changing decision-making models is evaluated based on three primary criteria: **safety**, **efficiency**, and **passenger comfort**. Each criterion is assessed using specific sub-parameters, as detailed below:

- **Safety-** The safety of a model is evaluated based on two key conditions: collision and distance headway (DH). The first condition, collision, occurs when the ego vehicle comes into contact with another vehicle. If no collision occurs, the second condition, distance headway, is assessed.

  Distance headway (DH) is calculated for the follower vehicle. The DH threshold is determined by the following formula:

  $$\text{DH threshold} = \text{Minimum time headway for safety} \times \text{follower vehicle speed}$$

where the minimum time headway is considered 2 seconds [8]

Safety Score Classification

The safety score is classified based on the safety margin available between the actual distance headway (DH) and the threshold distance headway. The classification is as follows:

Safety score	**Description**	**Distance Headway (DH)**
0	Collision / Imminent danger	$DH \leq$ DH threshold
0.5	Potential danger	DH threshold $< DH \leq 1.2 \times$ DH threshold
1	Safe	$DH > 1.2 \times$ DH threshold

Table 1: Safety score classification based on distance headway.

Methodology- The safety evaluation process utilizes the lane change output data obtained as XML files. It uses leader gap, original leader gap, follower gap, original follower gap, ego speed, and follower speed to compute minimum distance headways for original leader, leader, and follower vehicles taking minimum time headway as 2s. The measured distance headways are compared with the minimum distance headways to assign a safety score as per table 1 for each lane change. Even if one of the measured distance headways is lower than the threshold then the overall score for that lane change is 0. If one of the measured distance headways is close to the threshold of up to twenty percent then the overall score is 0.5. If all the measured distance headways are greater than 1.2 times the threshold, then an overall score of 1 is assigned to that lane change.

For the overall safety score of the individual simulation with multiple lane changes, the above-mentioned procedure is repeated, i.e. If the safety score of even one lane change is 0, then the overall safety score for that simulation is zero. If one of the lane changes is 0.5, the overall safety score is 0.5 for that simulation. If the scores of all lane changes in a simulation are 1, then the overall safety score of that simulation is 1.

Model Evaluation- Model safety is assessed by comparing the overall safety score of each model. Each model is run through at least twenty simulations and the safety score distribution across these repeated randomized simulations is collected. Based on this distribution, an overall safety score is assigned to each model.

- **Efficiency Evaluation-** Efficiency was assessed by analyzing the fuel consumption of the `Ego` vehicle during its lane-changing maneuver and subsequent highway exit. Fuel consumption, expressed in kilometers per liter (km/L), represents a critical measure of the vehicle's operational performance. This evaluation was conducted using data extracted from XML files generated during simulated vehicle operations for each model [9].

  The fuel consumption computation process utilized the tripinfo elements within the XML files, specifically focusing on entries where the vehicle ID was identified as `Ego`. Key parameters extracted included the total fuel consumption

(`fuel_abs`), route length (`routeLength`), and trip duration. The total fuel consumption, initially recorded in milligrams, was converted to liters for analysis. This conversion was achieved in two stages: (1) dividing the recorded fuel consumption by 1,000 to obtain the mass in grams and (2) dividing this value by the fuel density of 740 g/L to derive the volume in liters. Similarly, the route length, initially measured in meters, was converted to kilometers. Fuel efficiency, expressed as kilometers per liter (km/L), was then computed as the ratio of the route length to the total fuel consumed. This methodology ensures consistent and precise evaluation of fuel efficiency across varying traffic conditions and route complexities. It provides a robust framework for assessing the energy performance of the `Ego` vehicle and enables meaningful comparisons across different models.

Model Evaluation- The efficiency of the `Ego` vehicle was assessed across multiple vehicle models by comparing their respective fuel consumption metrics. Each model underwent a minimum of twenty simulation runs, during which the average fuel consumption in km/L was computed. These average values were used as benchmarks for performance evaluation.

The model exhibiting the highest average fuel efficiency (i.e., the lowest fuel consumption per kilometer) was deemed the most efficient. This rigorous evaluation framework allows for a comprehensive understanding of the energy performance differences among the tested models, facilitating informed decisions for model selection and optimization.

- **Passenger Comfort:** Passenger comfort is an essential factor in vehicle operation, particularly during lane-change maneuvers and highway exits. It is evaluated by analyzing the longitudinal jerk experienced by the `Ego` vehicle during these operations. Jerk is defined as the rate of change of acceleration over time, measured in m/s^3. The classification of jerk values is critical to understanding passenger comfort, as it directly impacts the smoothness of the vehicle's movement. According to the study in [10], jerk is categorized as follows:

Description	**Jerk Range (m/s^3)**
Comfortable	0.0 to 2.0 or 0.0 to -2.0
Acceptable	2.0 to 5.0 or -2.0 to -5.0
Harsh	> 5.0 or < -5.0

Table 2: Passenger comfort classification based on jerk ranges.

  - Methodology:- The calculation of jerk for the `Ego` vehicle was performed using XML data generated from the vehicle simulations. The key elements in the XML files were the `motionState` tags, which contained the time and acceleration values of the Ego vehicle at each simulation step. The first step in the process was to identify the vehicle's configuration and associated motion data by parsing the `actorConfig` and `vehicle` elements in the XML structure.

Once the Ego vehicle was identified, the acceleration values were retrieved from the `motionState` elements. Jerk was then computed using the following formula:

$$J = \frac{\Delta a}{\Delta t}$$

where Δa is the change in acceleration and Δt is the time interval between two consecutive motion states. The acceleration data were processed for consecutive time steps, and the jerk values were computed for each valid data entry where both Δa and Δt were non-zero. This computation provided a time series of jerk values that reflect the dynamic behavior of the Ego vehicle during the maneuver.

The jerk values obtained from the simulations were analyzed to identify the worst-case scenarios for both acceleration and deceleration. Specifically, the maximum positive jerk (indicating rapid acceleration) and the minimum negative jerk (indicating harsh deceleration) were of primary interest. These worst-case jerk values were recorded for each simulation to evaluate the overall comfort experienced by the passengers.

This methodology offers a robust framework for evaluating passenger comfort based on longitudinal jerk. By using a series of simulations and analyzing the worst-case jerk values, the study presents an in-depth comparison of vehicle models, allowing for the identification of the most comfortable configurations. The findings contribute valuable insights into vehicle design, particularly in terms of optimizing smoothness and reducing discomfort during critical driving maneuvers.

– Model Evaluation:- To assess the impact of jerk on passenger comfort across different models, the longitudinal jerk of the `Ego` vehicle was compared among multiple vehicle configurations. For each model, twenty simulations were conducted. For each simulation, the three highest positive jerk values (maximum jerk) and the three lowest negative jerk values (minimum jerk) were recorded. This methodology was adopted to capture a more representative sample of extreme jerk events, accounting for variability across simulations. By considering multiple maximum and minimum values rather than a single extreme value, the analysis ensures a more robust and comprehensive assessment of the vehicle's performance under worst-case acceleration and deceleration scenarios.

  The model with the lowest overall jerk values, particularly with regard to both maximum acceleration and harsh deceleration, was considered the most comfortable in terms of passenger experience. By comparing the worst-case jerk values across different vehicle models, this evaluation provides an objective measure of smoothness and comfort under consistent operating conditions.

  The analysis of jerk offers a holistic perspective on passenger comfort during dynamic driving scenarios, where elevated jerk values correspond to abrupt changes in vehicle speed that can lead to discomfort. A lower jerk

profile, particularly during critical maneuvers such as lane changes and highway exits, signifies smoother operation and contributes to an improved overall passenger experience.

3 Evaluation framework

The evaluation framework is structured around three core components: (1) scenario definition, (2) simulation execution, and (3) results analysis. Each component is designed to ensure a systematic, reproducible, and modular approach to evaluating model performance.

3.1 Scenario Definition

The traffic scenarios are defined using SUMO's XML-based configuration files. To build a scenario, we need to develop different XML files:

- Node and Edge File: Specifies the road dimensions and the starting and ending points.
- Routes File: Defines the 4-lane highway conditions, and lane speed limits. Define the EGO vehicle properties and other vehicles like cars, buses, and trucks used for simulation.
- Connect File: Specify the junction connections for the exit road.
- Config File: Outlines the output requirements from the SUMO simulator for the ego vehicle. In this case, we consider trajectories, trip info, and lane changes as outputs.

The files together make up a so-called *scenario* which aims to reflect realistic urban traffic conditions and specific testing requirements.

3.2 Simulation Execution

The simulations are executed using TraCI (Traffic Control Interface), which offers an extensive API for real-time interaction with the SUMO simulator. It is via this API that custom logic can be injected into the simulation, e.g. the actual LCDM logic. More concretely, the interface enables:

- **Dynamic** control of vehicle behaviours, including lane changes and vehicle dynamics.
- **Monitoring** and logging of vehicle states, such as speed, position, acceleration, and many more metrics.
- **Integration** of custom decision-making models implemented in Python.

TraCI provides the flexibility to adapt simulations to various experimental conditions that are defined programatically and applied exactly as needed. The entry point defines a given scenario, the models to be considered, and optional simulation configuration parameters. These parameters are sent to the main component of the framework, namely the *SimulationManager* class which manages the interaction between Python and SUMO, as well as runs a simulation given a model and a scenario. Key tasks include initializing the simulation, controlling vehicle behavior, and retrieving runtime data. An overview visualization of the data-flow can be seen in Figure 4.

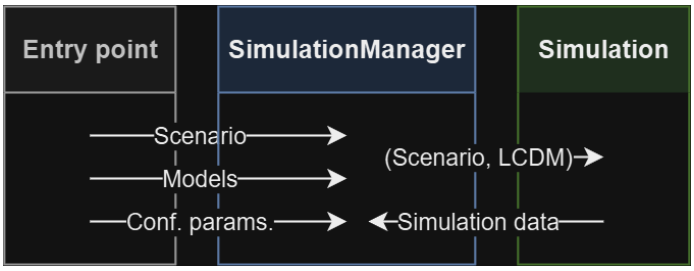

Figure 4: Data flow visualization

3.3 Results Analysis

The results of the simulations are output as XML files containing raw simulation data, which include detailed information about statistics such as traffic flow, vehicle states, and model performance metrics [11]. These data files are then further processed and analyzed using Python and frameworks such as *Numpy*, *Pandas*, and *Matplotlib*. The analysis focuses on key performance metrics such as lane-change frequency, collision rates, fuel efficiency, and travel time efficiency, providing insights into the performance of the decision-making models under different scenarios.

4 Results

The results presented here evaluate four vehicle models, each tested across 20 iterations with varying seed values to introduce randomness in traffic conditions. These models are as follows:

- Model A: Liu (Rule-Based Model)
- Model B: Machine Learning Model (SVM)
- Model C: Our enhancement of Liu (Rule-Based Model)
- Model D: SL2015

Each model's performance was assessed under diverse traffic conditions, providing a comprehensive evaluation of its behavior in a dynamic environment. The ego vehicle's task was to perform a lane change while traveling in a high-speed lane, moving to a slower-speed lane, and then exiting the highway. For all 20 iterations across each model, the vehicle successfully completed the lane change and highway exit task from the start of the simulation.
The simulation time for each model ranged from 300 ms to 800 ms per iteration, with the exact duration depending on factors such as traffic density and the specific scenario in which the model was tested. This variability underscores how different traffic conditions can influence the performance and responsiveness of each vehicle model, affecting their ability to execute maneuvers like lane changes and highway exits in a dynamic environment.

The demo video showcasing the ego vehicle successfully completing a lane change and making a highway exit for all models under Scenario C is available for viewing. You can access the video through the following link:
Demo Video for Scenario C

Following the presentation of the demo video, the performance of each vehicle model is evaluated in more detail under the different traffic scenarios: Scenario A, Scenario B, and Scenario C. The subsequent analysis will compare the models based on their behavior in these distinct traffic conditions, providing insights into their respective strengths and weaknesses.

4.1 Evaluation of the models' performance under Scenario A

Safety: Figure 5 illustrates the safety scores of the models for each iteration. The safety score is categorized as follows: A score of 0 indicates imminent danger/collision, 0.5 represents potential danger, and 1 signifies a safe condition. From the results, it can be observed that **Model D** is the least safe compared to the other models, as it contains only two iterations with a score of 0.5, and the remaining iterations have a score of 0. This suggests that the model consistently operates under unsafe conditions. On the other hand, **Model B** and **Model C** exhibit similar safety performance, with both models having four iterations with a score of 0.5, and the rest of the iterations having a score of 0. These models demonstrate a similar safety profile, indicating they generally operate under comparable conditions of potential danger.
Model A stands out with better safety performance. It includes two iterations with a score of 1 (safe) and the rest scoring 0. This suggests that **Model A** is the safest among the four models in some situations, although there is still room for further improvement. In conclusion, **Model A** proves to be the most reliable in terms of safety, outperforming the other models. However, continued improvements are necessary to achieve optimal safety in all iterations.

```
Model A safety score:  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0]
Model B safety score:  [0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0.5, 0, 0, 0, 0, 0.5, 0.5, 0, 0, 0, 0]
Model C safety score:  [0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0.5, 0, 0, 0, 0, 0.5, 0.5, 0, 0, 0, 0]
Model D safety score:  [0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0]
```

Figure 5: Safety evaluation for scenario A with models

Efficiency: Figure 6 presents the average fuel consumption of the Ego vehicle across various models. Model A exhibits the highest fuel efficiency, achieving a consumption rate of around 8 km/L, indicating superior performance in terms of distance traveled per unit of fuel. This is followed by model B, model C, and model D, which record a fuel consumption of 3.0 km/L.

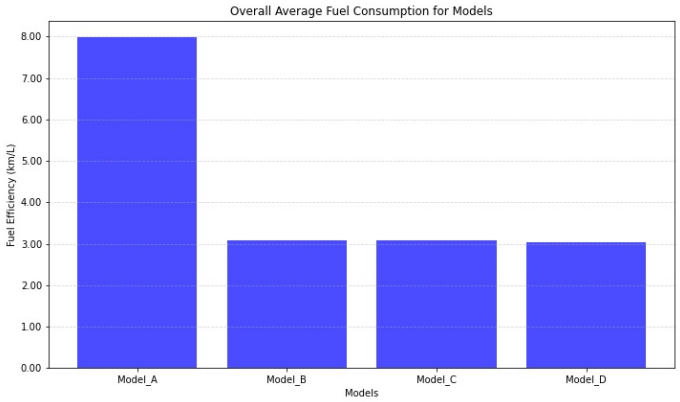

Figure 6: Efficiency evaluation for scenario A with models

Passenger Comfort: Figure 7 compares the worst maximum and minimum longitudinal jerk values across different models, with green bars representing maximum jerk (acceleration) and red bars representing minimum jerk (deceleration). For each simulation, the three highest positive jerk values and the three lowest negative jerk values were recorded to provide a more representative sample of extreme jerk events. This approach accounts for variability across simulations, ensuring a robust assessment of vehicle performance under worst-case acceleration and deceleration conditions.

Jerk values are measured in m/s³ and categorized into three thresholds based on passenger comfort: Comfortable (0.0 to ±2.0), Acceptable (±2.0 to ±5.0), and Harsh (greater than ±5.0). By considering multiple extreme values rather than a single outlier, the analysis offers a more comprehensive evaluation of ride quality.

Model D exhibits the most extreme deceleration jerk, with values of -6.81 and -5.06 m/s³, both categorized as **Harsh**, indicating significant discomfort during braking. In contrast, its acceleration jerk values remain within the **Acceptable** range, at 4.95 and 4.62 m/s³.

Model A's maximum jerk value of 5.86 m/s³ falls into the **Harsh** range for acceleration. However, its deceleration jerk values, ranging from -4.00 to -3.43 m/s³, are within the **Acceptable** range, suggesting smoother braking performance.

Models B and C show more consistent and balanced jerk profiles. Their maximum jerk values during acceleration are 4.86 and 4.78 m/s³, while their minimum jerk values during deceleration are -4.90 and -4.74 m/s³, respectively. All of these values fall within the **Acceptable** category, indicating a more comfortable ride in both acceleration and deceleration phases.

This comparison highlights that Models B and C provide the most comfortable passenger experience, as their jerk values remain within the tolerable limits in all scenarios. Conversely, Model D is the least comfortable due to its extreme deceleration jerk, which exceeds the **Harsh** threshold. The figure underscores the importance of managing jerk levels, particularly during deceleration, to improve passenger comfort.

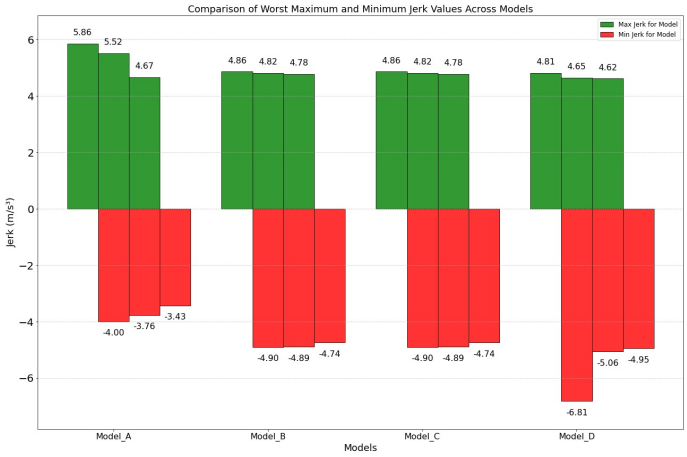

Figure 7: Passenger comfort evaluation for scenario A with models

4.2 Evaluation of the models' performance under Scenario B

Safety: Figure 8 illustrates the safety scores of the models for each iteration.
Based on the results, it can be observed that **Model A** is the least safe compared to the other models, as it contains only two iterations with a score of 0.5, with the remaining iterations having a score of 0. This suggests that the model consistently operates in unsafe conditions.
On the other hand, **Model B** and **Model C** exhibit similar safety performance, with both models having three iterations with a score of 0.5, and the rest of the iterations having a score of 0. These models demonstrate a similar safety profile, indicating they generally operate under comparable conditions of potential danger.
Model D stands out with a better safety performance. It includes one iteration with a score of 1 (safe) and three iterations with a score of 0.5, with the rest scoring 0. This suggests that **Model D** is the safest among the four models, though there is still room for further improvement. In conclusion, **Model D** proves to be the most reliable in terms of safety, outperforming the other models. Still, continued enhancements are necessary to achieve optimal safety across all iterations.

```
Model_A Safety Scores: [0, 0.5, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0, 0]
Model_B Safety Scores: [0, 0, 0, 0, 0, 0.5, 0, 0, 0.5, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0]
Model_C Safety Scores: [0, 0, 0, 0, 0, 0.5, 0, 0, 0.5, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0]
Model_D Safety Scores: [0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0.5, 0, 0.5, 0, 0, 0]
```

Figure 8: Safety evaluation for scenario B with models

Efficiency: Figure 9 presents the average fuel consumption of the Ego vehicle across various models. Model D exhibits the highest fuel efficiency, achieving a consumption

rate of 9.42 km/L, indicating superior performance in terms of distance traveled per unit of fuel. This is followed by Model A, which records a fuel consumption of 8.0 km/L. In contrast, Models B and C demonstrate significantly lower fuel efficiency, both with a consumption rate of 2.90 km/L. These findings underscore notable disparities in fuel efficiency among the models, with Model D outperforming all others, while Models B and C represent the least efficient configurations.

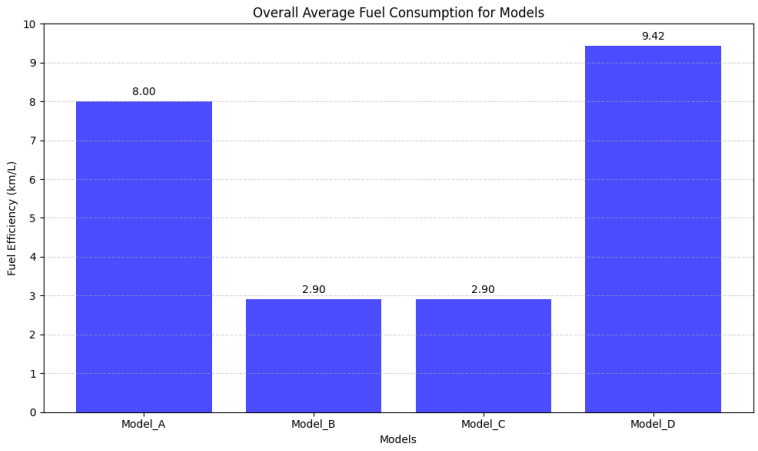

Figure 9: Effeciency evaluation for scenario B with models

Passenger Comfort: Figure 10 illustrates the passenger comfort of the ego vehicle across different models by comparing the three worst maximum and minimum longitudinal jerk values for each model. Jerk values are categorized into **Comfortable** (0.0 to ±2.0 m/s³), **Acceptable** (±2.0 to ±5.0 m/s³), and **Harsh** (greater than ±5.0 m/s³) to evaluate ride quality.

Model A consistently demonstrates the highest levels of discomfort, with all its extreme jerk values falling into the **Harsh** range. Its three worst maximum jerk values (7.0, 7.0, and 5.64 m/s³) indicate significant acceleration discomfort, while its minimum jerk values (-6.0, -5.0, and -5.0 m/s³) reflect equally harsh deceleration.

Models B and C perform better in terms of acceleration comfort, with their maximum jerk values (4.96, 4.95, and 4.80 m/s³) falling within the **Acceptable** range. However, their minimum jerk values (-6.75, -5.05, and -5.0 m/s³) fall into the **Harsh** category, highlighting discomfort during deceleration.

Model D provides a more balanced performance. Two of its maximum jerk values (4.88 and 4.76 m/s³) fall within the **Acceptable** range, while one value (7.0 m/s³) exceeds the threshold for harshness. Its minimum jerk values (-5.0, -4.62, and -4.58 m/s³) are comparable to those of Models B and C, remaining mostly within the **Harsh** range for deceleration.

Overall, Model A exhibits the lowest passenger comfort due to its consistently high jerk values, while Models B and C offer acceptable acceleration performance but harsh

deceleration. Model D strikes a balance but still features occasional harshness during acceleration and braking, emphasizing the need for improvement across all models.

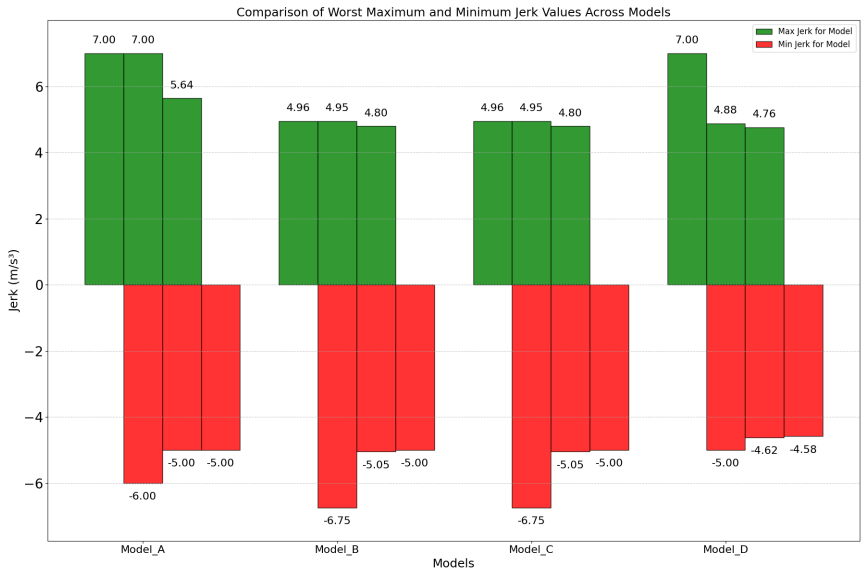

Figure 10: Passenger comfort evaluation for scenario B with models

4.3 Evaluation of the models' performance under Scenario C

Safety: Figure 11 illustrates the safety scores of the models for each iteration in the lane change scenario.

Based on the results, it can be observed that **Model A** is among the least safe, as it contains only three iterations with a score of 0.5, while the rest of the iterations have a score of 0. This indicates that the model operates predominantly in unsafe conditions, with very few iterations reaching potential safety. Similarly, **Model B** exhibits slightly better safety performance compared to Model A, with four iterations having a score of 0.5 and the remaining iterations scoring 0. This suggests that the model has a slightly higher likelihood of operating in potentially safe conditions but still falls short of achieving consistent safety. **Model C** shows comparable safety performance to Model B, with three iterations having a score of 0.5 and all other iterations scoring 0. This indicates that the model operates mostly in unsafe conditions, with only marginal improvements over Model A. On the other hand, **Model D** stands out as the safest model among the four. It includes one iteration with a score of 1 (safe), three iterations with a score of 0.5, and the remaining iterations scoring 0. This highlights that Model D achieves a better balance between safe and potentially safe conditions, though it still experiences unsafe iterations in a majority of cases.In conclusion, **Model D** emerges as the most reliable model in terms of safety, outperforming the others. However, further

improvements are required to enhance its safety consistency across all iterations.

```
Model A  safety score: [0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0.5]
Model B  safety score: [0.5, 0, 0, 0.5, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0, 0]
Model C  safety score: [0, 0, 0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 0]
Model D  safety score: [0.5, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 1, 0, 0.5]
```

Figure 11: Safety evaluation for scenario C with models

Efficiency: Figure 12 presents the average fuel consumption of the Ego vehicle across various models. The results indicate that all models—Model A, Model B, Model C, and Model D—achieve nearly identical fuel efficiency, with a rate of approximately 12.0 km/L. This demonstrates that, despite variations in safety scoring or other performance metrics, the fuel efficiency of the models remains consistently high and uniform across all configurations, showing no significant disparities in distance traveled per unit of fuel.

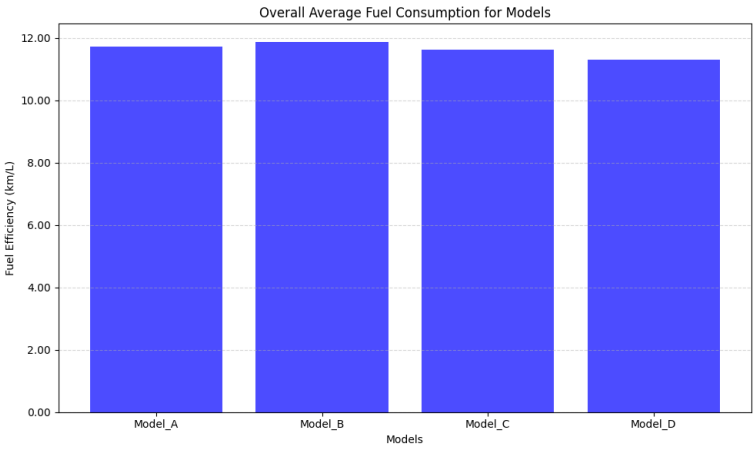

Figure 12: Fuel efficiency evaluation for scenario C

Passenger Comfort: Figure 13 illustrates the passenger comfort of the ego vehicle across different models by comparing the three worst maximum and minimum longitudinal jerk values for each model. Jerk values are categorized into Comfortable (0.0 to ±2.0 m/s³), Acceptable (±2.0 to ±5.0 m/s³), and Harsh (greater than ±5.0 m/s³) to evaluate ride quality.

Model A demonstrates high levels of discomfort, with its worst maximum jerk values reaching 6.66, 6.48, and 6.43 m/s³, all categorized as **Harsh**. Similarly, the worst

minimum jerk values for Model A, recorded as -6.43, -5.26, and -5.05 m/s³, also fall in the **Harsh** range, indicating rough accelerations and decelerations. Model B shows slightly improved performance with its worst maximum jerk values at 7.00, 5.91, and 5.53 m/s³, of which only one falls into the **Harsh** category, while the rest are borderline. However, the worst minimum jerk values for Model B, at -6.40, -5.42, and -5.15 m/s³, remain within the **Harsh** range, indicating harsh deceleration. Model C has the highest maximum jerk value of 7.10 m/s³, which is categorized as **Harsh**, with other maximum values of 6.82 and 6.59 m/s³, also in the **Harsh** range. Similarly, its worst minimum jerk values, at -7.04, -5.89, and -5.48 m/s³, are classified as **Harsh**, making it less comfortable overall. Model D, however, exhibits the best performance in terms of passenger comfort. Its worst maximum jerk values of 3.20, 2.65, and 2.63 m/s³ fall entirely within the **Acceptable** range, avoiding the harsh category. The minimum jerk values for Model D, recorded as -3.38, -3.20, and -2.97 m/s³, also fall comfortably within the **Acceptable** range, providing the smoothest experience among the models. In conclusion, Model D stands out as the most comfortable model for lane-change decision-making, with all jerk values remaining within the **Acceptable** range. Models B and C exhibit mixed results with a combination of acceptable and harsh jerk values, while Model A consistently produces harsh values, making it the least suitable for passenger comfort.

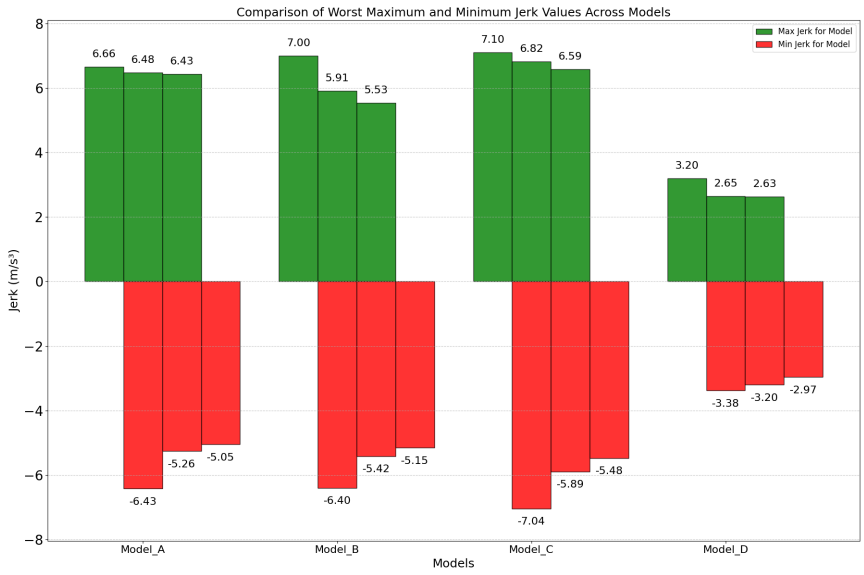

Figure 13: Passenger comfort evaluation for scenario C with models

The overall summary of all models across the three scenarios is provided in Table 4.3. The table presents the performance of each model in terms of **safety**, **efficiency**, and **passenger comfort** under varying conditions. The terms are defined as follows:

- **Safety (Overall Score):** The safety score is derived from an evaluation of collision occurrences and distance headway (DH). The score is classified as:
  - **0:** Collision or imminent danger observed.
  - **0.5:** Potential danger observed, with distance headway close to the threshold.
  - **1:** Safe, with all distance headways exceeding the safety threshold by at least 20%.
- **Efficiency (km/L):** Efficiency is measured as the fuel economy of the vehicle, expressed in kilometers per liter. Higher values indicate better fuel efficiency.
- **Passenger Comfort (Jerk):** Passenger comfort is assessed based on the rate of change of acceleration (jerk). The classification includes:
  - **Harsh:** High jerk values leading to discomfort during acceleration or deceleration.
  - **Mixed:** Moderate jerk values with occasional harsh instances.
  - **Acceptable:** Smooth and comfortable acceleration and braking.

This comprehensive comparison provides a detailed understanding of the strengths and limitations of each model, forming the basis for the conclusions in the subsequent section.

Scenario	Model	Safety (Overall Score)	Efficiency (km/L)	Passenger Comfort (Jerk)
A	A	0.5 (Potential danger observed)	8.0	Harsh: High jerk during acceleration and deceleration
	B	0.5 (Potential danger observed)	3.0	Mixed: Acceptable acceleration, harsh deceleration
	C	0.5 (Potential danger observed)	3.0	Mixed: Acceptable acceleration, harsh deceleration
	D	0 (Collision or imminent danger)	3.0	Balanced: Acceptable overall jerk
B	A	0 (Collision or imminent danger)	8.0	Harsh: High jerk for both acceleration and deceleration
	B	0.5 (Potential danger observed)	2.9	Mixed: Moderate jerk, harsh deceleration
	C	0.5 (Potential danger observed)	2.9	Mixed: Moderate jerk, harsh deceleration
	D	1 (Safe across all iterations)	9.42	Acceptable: Smooth acceleration and braking
C	A	0 (Collision or imminent danger)	12.0	Harsh: Extreme jerk in acceleration and deceleration
	B	0.5 (Potential danger observed)	12.0	Mixed: Acceptable acceleration, harsh deceleration
	C	0 (Collision or imminent danger)	12.0	Harsh: Extreme jerk in acceleration and deceleration
	D	1 (Safe across all iterations)	12.0	Acceptable: Smoothest across all scenarios

Table 3: Performance Summary of Models Across Scenarios A, B, and C Based on Safety, Efficiency, and Comfort

The results presented in Table 4.3 highlight both the improvements and trade-offs between the original Liu model (**Model A**) and our enhanced version of it. (**Model C**) across various scenarios. **Model C** shows improvements in passenger comfort by maintaining lower overall jerk values, as demonstrated in the results for Scenarios A and B. **Model A** consistently outperforms **Model C** in efficiency, achieving significantly better fuel consumption in Scenarios A and B, while in Scenario C, both models demonstrate a similar level of efficiency. In terms of safety, the results vary by scenario: **Model C** performs better in Scenario B, with fewer dangerous iterations, while **Model A** exhibits stronger safety performance in Scenario A and a similar level of safety performance in Scenario C, indicating comparable levels of security. These results emphasize the

trade-offs inherent in the design of decision models: while **Model C** offers greater adaptability to dynamic traffic conditions, this comes at the cost of higher fuel consumption and occasional inconsistencies in safety performance compared to **Model A**.

5 Conclusion

This study aimed to compare and evaluate lane-change decision-making (LCDM) models within the context of autonomous vehicles. The main focus was on their performance across metrics such as safety, efficiency, and passenger comfort metrics. By analyzing different LCDM models such as rule-based, machine learning-based, and hybrid models in varying traffic conditions using the SUMO simulation environment, we identified key strengths and limitations of each approach.

From a safety perspective, rule-based models naturally exhibit consistency in maintaining a moderate level of safety, although they struggle with adaptability to complex traffic dynamics. Machine learning models demonstrated potential for handling dynamic traffic but seemed less consistent from a safety perspective due to occasional erratic behavior. This behavior could stem from the original dataset used to train the model as the data collected was not explicitly made for the three scenarios but instead mapped on human behavior in dense traffic conditions. In turn, the model's accuracy does not transfer to the different criteria as it merely compares how the model performs when compared to the gathered dataset which does not necessarily have to be safe. Also, the high recall seen in Figure 2 can make the model more aggressive forcing the model to make more lane changes than necessary. Improved fine-tuning and optimization of the model parameters could have made the model perform better in the set scenarios. An interesting proposal is to create a new model comprised of both machine learning and rule-based to increase the safety aspect of the decision model. Hybrid models, particularly our enhancement of Liu model, achieved a better balance between safety and adaptability, though it requires granular refinement in order to be applicable in complex real-world scenarios. Nonetheless, hybrid models yield the flexibility of allowing generalized LCDM while still being interpretable as there's a rule-based foundation from which the machine learning part makes decisions. Fuel efficiency across the models revealed some disparities. The SL2015 model was most consistent in scoring high in the fuel efficiency metric, achieving higher **km/L** rates in moderate and high-traffic scenarios. Machine learning models tended to be less efficient, which ties in with what was previously mentioned regarding the need for additional optimization of the decision-making process to fine-tune without compromising performance in one or more aspects. Passenger comfort is an important aspect as lane changes typically are preferred to be smooth and comfortable rather than hectic and jerky. Longitudinal jerk analysis indicated that hybrid models as well as SL2015 provided smoother acceleration and deceleration profiles. However, like other rule-based approaches, it showed some limitations in adapting to highly complex traffic situations. Rule-based and purely machine-learning models tended to exhibit slightly harsher jerks during rapid decision-making. The main suspicion for these behaviors in

such models is that they require more granular fine-tuning than hybrid models.

The comparative analysis highlights that while rule-based models offer high interpretability and predictable behavior, they struggle in complex high-density traffic environments. Machine learning models excel in adaptability, but lack interpretability, which raises challenges from the perspective of requiring regulatory acceptance and deployment in safety-critical applications. Hybrid approaches, such as the our enhancement of Liu model, show promise by leveraging the strengths of both paradigms, though achieving optimal performance requires continued development. The study underlines the importance of evaluating LCDM models not only on individual metrics, but also on their holistic impact across safety, efficiency, comfort, and potentially more metrics. The use of SUMO as a simulation tool proved instrumental in testing and validating model performance under realistic traffic scenarios.

There is a lot of future work that can be done. While SUMO provided a controlled simulation environment, incorporating real-world traffic data could enhance the robustness and applicability of the models. Future research should also explore techniques to improve the interpretability of machine learning models, such as explainable AI methods. Due to the limited time available, not many traffic scenarios were created. Thus, future work should include a broader range of traffic scenarios, including urban and mixed-traffic conditions, which could provide deeper insights into their generalizability. Another interesting concept is the idea of allowing vehicles to communicate and share decision-making data with each other via some collaborative framework. Finally, the fine-tuning of the parameters for all the models as well as potentially introducing new models such as reinforcement learning-based approaches might lead to improved performance metrics. One way of potentially fine-tuning is to construct a Pareto front in order to find efficient choices with respect to given restrictions.

In conclusion, this study contributes to the ongoing development of reliable, efficient, and comfortable lane-change decision-making systems for autonomous vehicles. By emphasizing the trade-offs between interpretability, adaptability, and performance, the research provides a foundation for further advancements in intelligent transportation systems.

References

[1] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, and S. E. Li, "Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103 452, 2022, ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2021.103452. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X21004411.

[2] S. U. C. 2016, "Simulation framework for testing advanced driver assistance systems in chinese traffic situations," in *SUMO 2016 – Traffic, Mobility, and Logistics: Proceedings of the SUMO 2016 Conference*, M. Behrisch, M. Weber, and J. Erdmann, Eds., Chapter 11, Berlin, Germany: DLR – German Aerospace Center, 2016, pp. 103–113. [Online]. Available: https://elib.dlr.de/106342/1/SUMOconference_proceedings_2016.pdf.

[3] German Aerospace Center, *Sublanemodel - SUMO documentation*, Accessed 20 Jan. 2025, German Aerospace Center, 2025. [Online]. Available: sumo.dlr.de/docs/Simulation/SublaneModel.html.

[4] Liu, Yonggang, Wang, Xiao, Li, Liang, Cheng, Shuo, and Chen, Zheng, "A novel lane change decision-making model of autonomous vehicle based on support vector machine," *IEEE Access*, vol. 7, pp. 26 543–26 550, 2019. DOI: 10.1109/ACCESS.2019.2900416.

[5] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. DOI: 10.1007/bf00994018. [Online]. Available: https://doi.org/10.1007/bf00994018.

[6] D. Li and C. Ma, "Research on lane change prediction model based on gbdt," *Physica A: Statistical Mechanics and its Applications*, vol. 608, p. 128 290, 2022, ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2022.128290. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437122008482.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] F. Alonso, J. Naranjo, and F. Garcia, "An improved method to calculate the time-to-collision of two vehicles," *International Journal of Intelligent Transportation Systems Research*, vol. 11, Oct. 2011. DOI: 10.1007/s13177-012-0054-4.

[9] J. Vergés, "Analysis and simulation of traffic management actions for traffic emission reduction," Ph.D. dissertation, Jul. 2013.

[10] K. N. de Winkel, T. Irmak, R. Happee, and B. Shyrokau, "Standards for passenger comfort in automated vehicles: Acceleration and jerk," *Applied Ergonomics*, vol. 106, p. 103 881, 2023, ISSN: 0003-6870. DOI: https://doi.org/10.1016/j.apergo.2022.103881. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687022002046.

[11] DLR, *Sumo documentation: Simulation output*, Accessed: 2025-01-23, 2025. [Online]. Available: `https://sumo.dlr.de/docs/Simulation/Output/index.html`.