Towards Data Driven Traffic Modelling: Safe Driving Based on Reinforcement Learning*

Vasilis Kyriazopoulos, Foteini Orfanou, Eleni I. Vlahogianni, *Member, IEEE,* and George Yannis

***Abstract*— In recent years, the evolution of technology has allowed the introduction of automation in vehicles, that improve road safety by reducing the contribution of the human factor to the driving process. The objective of this research is to propose a reinforcement learning algorithm for controlling driving behaviour with the aim to improve safety and comfort. The learning is based on detailed trajectory data from a highly visited signalized arterial in the Athens downtown area. The safe and comfortable driving profiles are identified from the trajectory data. Next, a simple Q-learning algorithm is developed and various combinations of the exploration rate, the discount factor γ and the learning rate were tested for the optimal parameterization. The final Q-Table can be used inside vehicles for collision avoidance in order to improve road safety. Results indicate that the algorithm converges fast and is trained efficiently to response to unseen conditions. Further training in extreme events or adverse weather conditions will increase the generalisability of the proposed safe driving assistance framework.**

I. Introduction

There is a variety of different analytical behavioural models presented in the literature the last 25 years that have been customized to replicate autonomous vehicles and automated traffic conditions. Some examples include the Gipps model [1], the cellular automata [2], the Krauss car following model [3],[4], and the Intelligent Driver Model (IDM) [5]. Some recent studies use extensively the ACC [6] and the CACC models [7] that introduce connectivity in the process of driving.

Many researchers argue that models trained using real data can give better results in comparison to traditional models which are based on mathematical relations of traffic flow as well as those trained in simulated environments. The computing power of today's systems and the flexibility of machine learning algorithms make it possible to manage large volumes of data in real time to create models for various complex problems.

Recently, Reinforcement Learning (RL) methods have been used efficiently for configurating autonomous driving systems. RL provides the tools to solve and overcome problems that consist of multiple dimensions and an environment that cannot be described by a finite number of states are quite difficult to approach with standard methods. The target is developing autonomous vehicles that would resemble the complexity and sophistication of human (vehicle) decision-making processes. Modelling automated and autonomous vehicles (AV) should result in safe but also accepted driving profiles. Therefore, while transitioning to autonomous traffic, vehicles should behave in a similar way to humans, so that drivers can expect their actions and to be easily trusted by their users. The main concerns arisen and should be tackled is what can be defined as an accepted behaviour and what are these specific attributes the AV should entail for "resembling" to human driving.

The aim of this paper is to create a reinforcement learning algorithm that can be used inside cars as a drivers' assistance system that based on the traffic conditions around the examined vehicle, will be able to identify the best action in each timestep. By identifying the distance between the examined vehicle and the vehicles that are around the vehicle (front, back, left, right), as well as the speeds of those vehicles, the algorithm categorizes the state in which the driving environment is and based on that state it proposes the optimal action in order to avoid collision and improve road safety.

II. Data Driven Behavioural Traffic Models

Data driven models are more flexible and reveal new variables important for driver behaviour modelling that could not be detected through traditional models which are based on a specific functional form usually overparametrized to achieve a good fit to reality. Data driven models are trained using vehicle data, and are validated and calibrated using various machine learning techniques. These models have been used for enhancing existing models describing the car-following [8]-[11], the lane changing behaviour [12]-[16] or both behaviours [17]. Other applications relate to the development of Adaptive Cruise Control (ACC) system and autonomous driving [18]-[20].

A machine learning technique widely used in many applications in transportation engineering is Reinforcement Learning. Reinforcement Learning (RL) is a modelling framework, which deals with decision makers (agents) interacting with their dynamic environment, aiming to maximize a cumulative reward signal they receive. The environment comprises everything outside the agent. A reinforcement learning problem is often conceptualized as a Markov Decision Process (MDP), which is a mathematical formalization of sequential decision making (Puterman, 1994). The growing body of literature dedicated to such approaches includes applications of simple Q-learning algorithms [21]-[23] to much more complex deep RL

Vasilis Kyriazopoulos, Foteini Orfanou, Eleni I. Vlahogianni and George Yannis are with the National Technical University of Athens, School of Civil Engineering, 15773, Athens, GR (e-mails:vaskiriazo@gmail.com; forfanou;elenivl;geyannis@central.ntua.gr).

structures, such as the double deep Q-network (DDQN) algorithm [24], the deep deterministic policy gradient (DDPG)[25].

Recently, Zhu et al. [26] developed a deep deterministic policy gradient that can reproduce human behaviour more accurately than basic recent models, including the smart driver model, locally weighted regression models and conventional neural networks. The research concludes that reinforcement learning can be used in creating autonomous driving algorithms that mimic human behaviour, while they are particularly good for generalization.

Most of the research studies implementing reinforcement learning techniques for modelling the behaviour of AVs, use the spatial or temporal distance between the examined and the preceding vehicle as indicator for the model development and performance evaluation. The present work extends past research by using the distances in both lateral and longitudinal axes, i.e by taking into consideration the existence of vehicles around the examined one 9front, back, left, right) and the corresponding distances. These values are compared to the thresholds identified in the literature for critical driving conditions and are the basis for determining what kind of actions the agent should apply in order to improve the safety levels.

III. The Model

A simple Q learning algorithm was chosen for being developed in the present work. In this framework, at each time step t, the optimal policy $\pi*$ derives from $\pi* = argmax_{a\in A}\, Q*(s,a)$, while Q*(s,a) is defined as:

$$Q*(s,a) = \sum_{s'\in S} p(s'|s,a)\left(R_a(s,s') + \gamma \max_{a'\in A} Q*(s',a')\right) \quad (1)$$

where $\gamma \in [0,1)$ is the discount factor, representing the likelihood to reach state s' from state s, through a set of actions. Dynamic programming (e.g. value or policy iteration algorithms) is a typical way of computing optimal Q-values while model-free approaches can be favoured when the transition probability functions and reward values are unknown. The Q-learning algorithm (Watkins, 1989) is a standard technique that learns a Q-value function by executing actions in an environment and observing rewards. If the agent is currently in state s, action a is executed to transition to another state s' and reward r is observed, then the following rule is used to update the estimate of $Q(s,a)$:

$$Q(s,a) := Q(s,a) + a_q\left(r + \gamma \max_{a'\in A} Q(s',a') - Q(s,a)\right) \quad (2)$$

where $a_q \in [0,1]$ is the learning rate and γ the discount factor. The learning rate, alpha ($a_q \in [0,1]$), controls the difference between previous and new estimate of Q-value and is expected to initially receive a high value, allowing fast changes, and to decrease as time progresses due to concurrent improvement in agent performance. The factor gamma ($\gamma \in [0,1)$), determines the importance of future rewards. A factor approaching 1 will cause taking into account the long-term consequences while a value equal to 0 will render the agent short-sighted by only considering immediate rewards, while. When the Q-values have nearly converged to their optimal values, the agent is opting for the action with the highest Q-value. Parameter ε defines the exploration mechanism in ε-greedy action selection, according to which the agent explores the environment by selecting an action at random with probability ε, or, alternatively, exploits its current knowledge by choosing the optimal action with probability 1–ε.

The selection of the Q-learning algorithm was based on the fact that the algorithm is exploration insensitive [27]. The way the agent behaves does not influence the convergence of the Q values to the optimal ones providing that all state-action pairs occur with satisfactory frequency. Although the exploration-exploitation issue must be addressed in Q-learning, the details of the exploration strategy will not affect the convergence of the learning algorithm. Yet, in the proposed framework, not only finite, but also small set of states and actions are formulated, justifying the selection of the Q-learning algorithm.

IV. Application

A. The Data

The rapid development of unmanned aerial vehicles during the past few years has revolutionized the way traffic data is collected and processed. The data for the research were obtained from an experiment conducted in Athens (Greece) downtown area using drones **Error! Reference source not found.**. The purpose of the experiment was to record detailed trajectories in an urban environment and provide an overall picture of how the specific features of UAVs can overcome existing constraints in collected data using video recording from stationary cameras [29],[30]. The complete dataset can be accessed at: https://open-traffic.epfl.ch/.

The data used in the present analysis were collected on a major arterial in the centre of Athens with 3 lanes per direction and dense traffic especially during the peak hours. The data used were recorded every 0.8s for 30 minutes and the vehicle composition consisted of the following types: motorcycle, passenger car, taxi, van, truck and bus. Finally, the dataset included information concerning the vehicle coordination (x, y), its speed, its tangent and lateral acceleration and the corresponding timestep.

B. Driving States Identification

Most control problems cannot be represented by a specific number of states. This is mainly due to the "Curse of Dimensionality", where the dimensions of a problem are infinite. For example, in the driving problem there are different distances between the vehicle and the surrounding traffic, various speeds and positions of the vehicles as well as angles of movement. All these variables are continuous, resulting in an infinite number of states. In addition, the Markov property has to be taken into account in reinforcement learning. In order to simplify the problem of state representation while meeting all the above requirements the following factors are considered: i. the time to collision (TTC) between the examined and the vehicle in the front, ii. the TTC between the examined and the vehicle in the back, iii. the lateral distances between the examined and the surrounding vehicles. The term refers to the time required, usually in seconds, until two vehicles collide while maintaining fixed speeds. Provided that the algorithm will

provide guidance to a driver through messages, the reaction time of the driver should be taken into account.

C. Defining Critical Driving Conditions

To identify high risk (critical) driving conditions, the boundary values of TTC (plus reaction time) and lateral distances should be defined. The critical value of TTC has been the subject of extensive research with results not converging. Hist and Graham [31] report that a four second time measure could be used to distinguish between cases where drivers are inadvertently in a dangerous state and those where the drivers retain control. The study also describes an experiment to design a Collision Avoidance System. The results show that a four or five second collision warning criterion led to many false alarms. A time value of three seconds produced the minimum number of false alarms.

Hogema and Janssen [32] investigated behaviour in an unsupported and a supported driver approach during a driving simulation. The experiment yielded a minimum crash time of 3.5 seconds for unsupported drivers and 2.4 seconds for supported drivers. A value of 2.4 is considered critical in this research. Finally, VanDerHorst [33] reports even lower critical values for TTC. Based on the above a critical value for the TTC of 2.4 seconds is considered. This value is relatively conservative but improves overall safety. When a TTC lower than 2.4 seconds is present, the state will be deemed dangerous.

Most research estimate the reaction time based on experiment conducted in a simulator. Johansson and Rumar [34], examined the reaction time and concluded that values range from 0.4 to 2.7 secs. However, since drivers knew that they participated in an experiment to calculate the reaction time and a trigger sound was used, the values may not be objective. Another recent research using a combination of simulation and real-world driving has shown that the reaction time of average driver to an expected real-world event is 0.42 secs [35]. At the same time, the average reaction time in an unexpected danger is 1.1 seconds and in a simulator 0.9. When investigating reaction time, Lerner [36] did not inform drivers that they were involved in reaction time research and their reaction times range between 0.7 and 2.5 secs.

The reaction value that is used in this research is equal to 1.5 seconds, which represents most drivers' behavior. This should be added to the TTC, so that the driver receives the message and manages to react before the TTC becomes critical. Thus, the time to collision to alert the driver of a potentially dangerous state will be 3.9 seconds.

The critical lateral distance is defined as the distance between two vehicles, beyond which there is a high risk of lateral collision. Critical lateral distances are calculated for combinations of cars with other types of vehicles. The 4 categories of vehicles considered are: i. Car, ii. Bus, iii. Semi-truck, iv. Two wheeler.

The critical lateral distances are calculated based on distribution of all lateral distances of the combinations observed in the data. The distributions approach the normal distribution and are therefore considered to be normal values. The lateral distances on the left and the right side of vehicles are identical, so the critical distances between the examined and the vehicles on the left and right are considered equal. The lateral distances end up being similar because the left distance of the middle vehicle represents the right distance of the left vehicle and the right distance of the middle vehicle represents the left distance of the right vehicle. The critical distances are derived from the mean value of the distribution by subtracting the standard deviation. These values account for 84.14% of the distribution, that is, 15.86% of vehicles maintain a critical lateral distance. This percentage was conservatively chosen to account for the variation in vehicle width, as vehicle sizes vary and available measurements are center-to-center of vehicles and not pure lateral distances. Based on this percentage the critical distances are summarized in Table I.

TABLE I. MEASURED CRITICAL LATERAL DISTANCES FOR DIFFERENT VEHICLE TO VEHICLE INTERACTIONS.

Interactions	Critical Distance (m)
Car – Two wheeler	1.359
Car-Car	1.925
Car-Semi-truck	2.382
Car-Truck	2.556
Car-Bus	2.466

D. State Identification through Machine Learning

The states are defined in two stages. In the first stage, longitudinal stimuli for emerging driving patterns are considered, namely the vehicles in front of and behind the examined vehicle. Figure 1 shows the first stage of determining the state of the environment. If there is a vehicle in front of the examined vehicle, then the Time to Collision with the front vehicle is considered (TTCF). Then, if there is a rear vehicle, the Time to Collision with the back vehicle (TTCB) is examined in both cases the TTC is considered critical when its value is less than 3.9 seconds. Based on the above criteria, nine levels of separation occur.

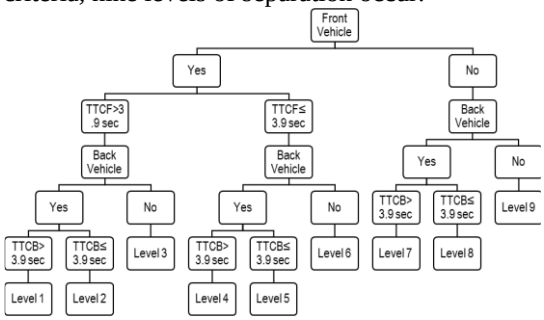

Figure 1. First stage of state determination (longitudinal direction)

During the second stage (Fig. 2), the lateral stimuli coming from the vehicles on the left and the right side of the subject vehicle are considered. We distinguish between the following cases: i. No lateral vehicles, ii. One lateral vehicle and iii. Two lateral vehicles. If there are more than one vehicles on one side of the examined vehicle, the one with the smaller lateral distance is considered. When lateral vehicles are present, the criticality of the distance is examined. This results in six levels of separation. In the case where there is one side vehicle at a critical distance, it should be specified on which side the critical vehicle is located, so at last 8 levels occur.

In total, the environment can be represented by 72 states, as each of the nine levels of stage 1 can be further divided into eight levels based on stage 2.

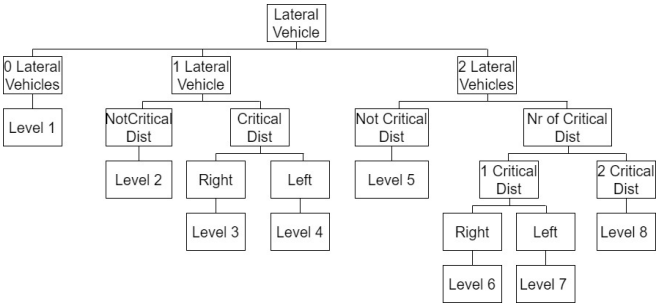

Figure 2. Second stage of state determination (lateral direction)

E. Identification of Actions

The implementation of the algorithm requires the identification of available actions that the driver can select. These occur through the driving process and are distinguished by the joint consideration of steering and acceleration. The vehicle can accelerate/decelerate and be turned left or right relative to its current direction. In any case, there is also the possibility of no action being taken, the vehicle continues at the same speed and in the same direction. Based on the above 9 actions can be distinguished, as shown in Table II. The long-term goal of the driver is to discover the best actions for each state and, thus, maximize his cumulative reward.

TABLE II. IDENTIFIED ACTIONS.

Actions	
Cruising (No change in speed)	Turn Left – Acceleration
Acceleration	Turn Left – Deceleration
Deceleration	Turn Right – Acceleration
Turn Left – No change in speed	Turn Right – Deceleration
Turn Right – No change in speed	

F. Reward Function

After the agent selects an action, the reward function evaluates this option and returns a reward. This evaluation is carried out as follows: If the agent chooses the appropriate action that leads the vehicle to a safer state, the reward is one unit. Half a unit for the one axis of movement (front – rear) and half a unit for the other axis (right – left). If the agent chooses the appropriate action for one axis and a wrong action for the other, the reward is minus half a unit as the agent is guided to the right policy without receiving the maximum possible reward. The agent receives half a unit for the correct choice on one axis and minus one unit for the wrong choice on the other axis. The overall reward is negative as the agent's action is still wrong and does not improve safety. If the agent chooses the wrong actions on both axes, the reward is minus two units. This is because the action leads the vehicle to a potentially more dangerous state than the initial one.

The correct action in every state is based on logic. When there is a critical TTCF, the driver should reduce speed, while when there is a critical TTCB, the driver should increase speed, which despite going against human nature, is the appropriate action. When there is a vehicle at a critical distance to the left of the examined vehicle then the vehicle should turn right, while when there is one on the right, it should turn left. During extreme cases, the driver should remain inactive.

V. RESULTS

To find the optimal algorithm, various combinations of the parameters were tested. The three parameters of Q-learning are the learning rate lr, the discount factor γ and the exploration rate e. The range of these parameters is from zero to one. Six scenarios are evaluated as depicted in Table III.

TABLE III. Q-LEARNING ARLGORITHM PARAMETERIZATION.

Scenarios	**Parameters' values**
I	γ=0.9 lr=0.2 e=0.4
II	γ=0.9 lr=0.5 e=0.4
III	γ=0.9 lr=0.2 e=0.2
IV	γ=0.9 lr=0.2 e=0.5
V	γ=0.9 lr=0.2 e=0.7
VI	γ=0.7 lr=0.2 e=0.2

The learning rate determines to what extent the new knowledge outweighs the old. A higher learning rate leads the algorithm to converge faster than a lower rate. The discount factor γ reflects how important future rewards are. Small values of γ give more weight to present rewards while high values lead to future rewards being considered. The exploration rate represents the percentage of time steps in which the agent chooses random actions. By using a random number generating function, an action is selected ignoring the policy.

A. Convergence

It is important for the diagram of the mean optimal values of all states to converge after many iterations. This is an indication that the algorithm has completed its training and can be used. Also, it is important for the convergence to happen fast. During testing all parameter combinations lead to a diagram that managed to converge relatively fast. The diagram of the third scenario is included for reference in Figure 3.

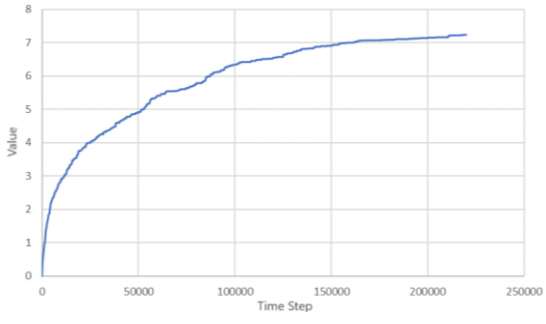

Figure 3. Value of the mean optimal action (γ=0.9, lr=0.2, e=0.2)

B. Regret

An additional way to evaluate the results is to compare the regret for the different parameter combinations. Regret of a π algorithm is the difference between the cumulative reward of the optimal policy and the cumulative reward of the policy π

that was followed (Figure 4). The regret of policy π after T time steps is calculated as:

$$R^{\pi}(T) = TV_H^{\pi^*} - \sum_{t=1}^{T} \mathbb{E}[V_H^{\pi t}] \quad (3)$$

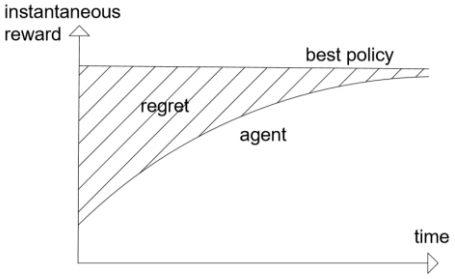

Figure 4. Definition of regret (source: Proutiere et al. 2019)

The calculation of the regret is done after 220000 steps for each scenario, when training ends. The maximum cumulative reward that the algorithm can receive in each scenario is 220000 units. Since the maximum reward during each time step is 1 according to the reward function. The final cumulative rewards for each scenario are seen in Table IV.

Scenarios 4 and 5 present a particularly high regret because of the use of a high exploration rate, so many of the agent's actions are chosen at random rather than based on the policy. Scenarios 3 and 6 appear to have the best regret values due to the fact that the exploration rate is low allowing the algorithm to follow mainly its policy without exploring new actions.

TABLE IV. RESUTS FOR EACH SCENARIO AFTER CONVERGENCE.

Scenario	Cummulative reward	Regret
I	43233.5	156766.5
II	43175	156825
III	131490.5	68509.5
IV	-425.5	200425.5
V	-87961	287961
VI	131507	68493

C. Accuracy

The accuracy of the results of each scenario is a good indicator of the quality of every parameter combination. While calculating the accuracy, the trained algorithm is tested using the evaluation data (a set of data that the algorithm has not been exposed to prior to the testing), where actions are chosen and rewards are received depending on the quality of the actions. The evaluation data consist of 54976 instances. By calculating in how many of these iterations the agent chose the wrong action, conclusions about the quality of the algorithm can be derived. For each scenario the wrong choices are seen in Table V.

TABLE V. ACCURACY FOR EACH SCENARIO.

Scenario	Wrong choices
I	10
II	13
III	4
IV	6
V	10
VI	9

Generally, all scenarios have a small number of errors since they were examined at 54976 instances. Regardless of the parameters, all scenarios had the opportunity to learn from a large number of examples. However, an algorithm intended as a driver's assistance system could not be accepted with a relatively large number of errors. Accuracy is extremely important when it comes to such systems.

D. Optimum Parameterization and Modelling Implications

The final algorithm is the one with parameters of scenario 3 as this combination had the least errors, a small regret and converges. The value of γ=0.9 indicates that the algorithm relies heavily on future rewards while the e=0.2 allows the algorithm to explore without leading to random selection of actions for a large number of repetitions and finally the lr=0.2 is a standard value so that knowledge already acquired is not ignored for the sake of new knowledge. Based on the above, the optimal Q-Table is obtained, which contains the values for all combinations of states and actions. The table allows the selection of the optimal action for each state in which the environment can be found. Some states have zero values for all actions. This is because the algorithm was not exposed to these states during training. Moreover, some zero values in actions are the result of the algorithm not exploring these states sufficiently. Shortcomings are mainly in some extreme cases. This problem can be attributed to the fact that although the sample that was used during training is large, it does not contain many extreme cases. During the data collection, no collisions or marginal collisions were observed so that the algorithm could learn to behave in particularly dangerous states. The Q-Table can be used as a driver assistance system, where it will provide signals while driving to avoid collisions with other vehicles. As part of the transition to autonomous traffic, the algorithm could be part of an autonomous driving system, since it is able to manage the interaction of the vehicle with the rest of the traffic, to identify any dangerous states and take appropriate actions.

VI. CONCLUSIONS

In this work, we proposed a simple reinforcement learning framework for improving driving behaviour. For this purpose, an agent was trained in order to be able to choose the optimal, between nine, action based on the state of the environment (72 states) and improve road safety. Findings reveal that the algorithm has the ability to learn quickly as the diagram of the value of the mean maximum action of all states converges. A reinforcement learning algorithm, in order to be accepted and used, must achieve convergence during training. The speed with which it reaches convergence is an indication of its quality. Therefore, the algorithm developed can be considered satisfactory as it converges rapidly.

All reinforcement learning algorithms have the ability to continue learning and improve without the need to create a new algorithm and start the training from the beginning compared to statistical models. Thus, the algorithm developed can be further trained to better deal with critical states. The algorithm is able to improve driving behaviour, since based on the actual data of the drivers, the model would avoid critical states. Reinforcement learning is an appropriate method for addressing issues such as the improvement of road safety by creating algorithms for driver assistance systems and autonomous driving systems. Next research steps could

be focused on comparing this simple Q-learning model with a deep reinforcement learning structure to understand whether feeding the model with the full range of the variables values and not just estimations can result in a more realistic representation. Furthermore, the drivers around the vehicle, and the driver of the vehicle could be categorized, based on their driving behaviours (normal, conservative, aggressive) which could help with the overall road safety by anticipating certain circumstances and proposing actions based on the characteristics of the drivers involved. This will contribute not only to safe driving behaviour of the "machine", but also in accepted reactions in non-critical and extreme events. Finally, more variables could be introduced, like weather, visibility and traffic conditions leading to more precise results and a more accurate version of the driving process.

ACKNOWLEDGMENT

The analysis is conducted within the framework of Drive2theFuture project (Needs, wants and behavior of "Drivers" and automated vehicle users today and into the future" funded by European Commission under the MG-3.3.2018: "Driver" behavior and acceptance of connected, cooperative and automated transport; Research and Innovation Action (RIA).

REFERENCES

[1] P. G. Gipps, "A model for the structure of lane-changing decisions." Transportation Research Part B: Methodological 20.5 (1986): 403-414.

[2] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic. Journal de physique I, 2(12), 2221-2229, 1992.

[3] S. Krauß, "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics", Dissertation. DLR-Forschungsbericht. 98-08, 115 S, 1998.

[4] S. Krauß, P. Wagner and C. Gawron, "Metastable states in a microscopic model of traffic flow", Physical Review E. 5. 5597-5602.

[5] M. Treiber, A. Hennecke and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations", Physical review E, 62(2), 1805, 2000.

[6] V. Milanés and S. E. Shladover, "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data", Transportation Research Part C: Emerging Technologies, 48, pp. 285–300, 2014.

[7] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput", Transportation Research Part C: Emerging Technologies, 71, 143-163, 2016.

[8] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu and C. Chen, "Data-driven intelligent transportation systems: A survey", IEEE Transactions on Intelligent Transportation Systems, 12(4), 1624-1639, 2011.

[9] H. Zheng, Y.-J. Son, Y.-C. Chiu, L. Head, Y. Feng, H. Xi, S. Kim and M. Hickman, "A primer for agent-based simulation and modelling in transportation applications". U.S. Department of Transportation - Federal Highway Administration. Report FHWA-HRT13-054, 2013.

[10] L. Chong, M. M. Abbas, A. M. Flintsch and B. Higgs, "A rule-based neural network approach to model driver naturalistic behavior in traffic", Transportation Research Part C: Emerging Technologies, 32, 207-223, 2013.

[11] C. Colombaroni and G. Fusco, "Artificial neural network models for car following: experimental analysis and calibration issues", Journal of Intelligent Transportation Systems, 18(1), 5-16, 2014.

[12] C. Ding, W. Wang, X. Wang and M. Baumann, "A neural network model for driver's lane-changing trajectory prediction in urban traffic flow", Mathematical Problems in Engineering, 2013.

[13] H. Bi, T. Mao, Z. Wang and Z. Deng, "A data-driven model for lane-changing in traffic simulation", In Symposium on Computer Animation, 149-158, 2016.

[14] Y. Hou, P. Edara and C. Sun, "Modeling mandatory lane changing using Bayes classifier and decision trees", IEEE Transactions on Intelligent Transportation Systems, 15(2), 647-655, 2013.

[15] P. Kumar, M. Perrollaz, S. Lefevre and C. Laugier, „Learning-based approach for online lane change intention prediction", In 2013 IEEE Intelligent Vehicles Symposium (IV) (pp. 797-802). IEEE.

[16] E. G. Wang, J. Sun, S. Jiang and F. Li, "Modeling the various merging behaviors at expressway on-ramp bottlenecks using support vector machine models", Transportation research procedia, 25, 1327-1341, 2017.

[17] L. Huang, H. Guo, R. Zhang, D. Zhao, J. Wu, "A data-driven operational integrated driving behavioral model on highways. Neural Computing and Applications, 1-17, 2020.

[18] Y. Tian, K. Pei, S. Jana and B. Ray, "DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars".

[19] X. Shi, Y. Wong, C. Chai and M. Li, "An Automated Machine Learning (AutoML) Method of Risk Prediction for Decision-Making of Autonomous Vehicles",IEEE Transactions on Intelligent Transportation Systems. PP. 1-10, 2020.

[20] F. Simonelli, G. N. Bifulco, V. De Martinis and V. Punzo, "Human-like adaptive cruise control systems through a learning machine approach", In Applications of Soft Computing (pp. 240-249). Springer, Berlin, Heidelberg, 2009.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M.G. Bellemare, ... and S. Petersen, "Human-level control through deep reinforcement learning", Nature, 518(7540), 529, 2015.

[22] A. Yu, R. Palefsky-Smith and R. Bedi, "Deep reinforcement learning for simulated autonomous vehicle control", Course Project Reports: Winter, 1-7, 2016.

[23] D. M. Vlachogiannis, E. I. Vlahogianni and J. Golias, "A reinforcement learning model for personalized driving policies identification. International journal of transportation science and technology, 9(4), 299-308. 2020.

[24] S. Nageshrao, H. E. Tseng and D. Filev, "Autonomous Highway Driving using Deep Reinforcement Learning," IEEE International Conference on Systems, Man and Cybernetics, Bari, Italy, 2019, pp. 2326-2331.

[25] M. Zhou, Y. Yu and X. Qu, "Development of an Efficient Driving Strategy for Connected and Automated Vehicles at Signalized Intersections: A Reinforcement Learning Approach," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 1, pp. 433-443, Jan. 2020, doi: 10.1109/TITS.2019.2942014.

[26] Zhu, M., Wang, X., & Wang, Y. (2018). Human-like autonomous car-following model with deep reinforcement learning. Transportation research part C: emerging technologies, 97, 348-368.

[27] L.P. Kaelbling, M.L. Littman, A.W. Moore, "Reinforcement learning: a survey", Learning (1996), pp. 237-285.

[28] E. Barmpounakis and N. Geroliminis, "On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment", Transportation Research Part C: Emerging Technologies, Volume 111, 2020, Pages 50-71.

[29] E. N. Barmpounakis, E. I. Vlahogianni and J. C. Golias, "Unmanned Aerial Aircraft Systems for transportation engineering: Current practice and future challenges", International Journal of Transportation Science and Technology, 5(3), 111-122, 2016.

[30] E. N. Barmpounakis, E. I. Vlahogianni, J. C. Golias and A. Babinec,. "How accurate are small drones for measuring microscopic traffic parameters?", Transportation Letters, 11(6), 332-340, 2019.

[31] Hirst, S., Graham, R., 1997. The format and presentation of collision warnings. In: Noy, I.Y. (Ed.), Ergonomics and Safety of Intelligent Driver Interfaces. Lawrence Erlbaum, Mahwah,

[32] Hogema, J. H., and W. H. Janssen. Effects of intelligent cruise control on driving behaviour: a simulator study. No. 1996 C012. TNO, 1996.

[33] A. R. A. Van der Horst, "A time-based analysis of road user behaviour in normal and critical encounters", 1991.

[34] NJ.G. Johansson and K. Rumar, "Drivers' brake reaction times", Human factors, 13(1), 23-27, 1971.

[35] T. Magister, R. Krulec, M. Batista and L. Bogdanovic, "Measuring a Driver's Reaction Time". Strojniski Vestnik, 52(1), 26-40, 2005.

[36] N. D. Lerner, R. W. Huey, H. W. McGee and A. Sullivan, "Older Driver Perception-Reaction Time for Intersection Sight Distance and Object Detection", Volume I: Final Report. Report No. FHWA-RD-93-168, Federal Highway Administration, 1995.