The Need for Location-based Machine Learning Models for Level 5 Automated Vehicles

Kaushik Madala and Hyunsook Do
Department of Computer Science and Engineering, University of North Texas, Denton, U.S.A.

Keywords: Safety Of The Intended Functionality (SOTIF), ISO 21448, ISO 26262, Machine Learning Safety, Autonomous Vehicle Safety, SAE Level 5 of Driving Automation.

Abstract: Assuring safety of machine learning (ML) models in autonomous vehicles is a challenging task. This is because of the complex operational design domain (ODD) settings under which we need to validate ML models. In particular, autonomous vehicles with level 5 of driving automation need to operate under any ODD conditions, and should ensure safety of both road users and passengers. However, deploying common ML models across a fleet of vehicles to operate in multiple regions can complicate the safety assurance process.Even when an ML model is found to be causing a crash due to an ODD condition occurring in only one of the regions, we still should update it across the fleets of all regions. If we can limit its update within that particular region, we can reduce the complexity of safety assurance. In this paper, we propose the location-based machine learning models for level 5 automated vehicles to address this problem and how they will be helpful compared to deploying instances of global ML models which are same across a company's fleets of vehicles.

1 INTRODUCTION

Automated vehicles are one of the highly complex systems that rely on machine learning (ML) models to perform tasks such as perception and motion planning (Chen et al., 2015; Aradi, 2020; Pfeiffer et al., 2017). The widely used machine learning algorithms are deep neural networks (Grigorescu et al., 2020). Some of the machine learning models generated using deep neural networks are object detection models (used to classify and locate objects), reinforcement learning models (used to predict a potential next state based on current state), and adversarial learning models (used to analyze robustness of other ML models). While the current state-of-the-art autonomous vehicles still rely on safety-drivers or remote operators to ensure safety and prevent accidents or operate in a limited geographical location, the ultimate goal is to reach SAE level 5 of driving automation (Committee et al., 2018), where the vehicle can autonomously travel to any place and ensure safety of passengers and other road users (e.g., other vehicles, pedestrians) without human feedback.

Performing safety assurance for autonomous vehicles with level 5 of automation can be very challenging. This is because of the extensive operation design domain (ODD) (BSI/PAS, 2020) that we need to consider for such vehicles. ODD includes factors such as environmental conditions (e.g., snowy weather), road users (e.g., pedestrian, bicyclists), and scenery/road infrastructure (e.g., pavement, road markings, road type, traffic signs). To ensure safety of autonomous vehicles, they should comply with the safety standards: ISO 26262 (ISO, 2018) and ISO 21448 (ISO/PAS, 2020). While ISO 26262 deals with functional safety (FuSa), ISO 21448 deals with safety of the intended functionality (SOTIF).

The ML models we use in an autonomous vehicle also need to comply with the ISO 26262 and ISO 21448 standards. In the case of an autonomous vehicle with level 5 of driving automation, compliance with ISO 26262 and ISO 21448 requires to sufficiently demonstrate that the vehicle has an acceptable or reasonable level of risk under all ODD settings. If we assume that the autonomous vehicles of level 5 are being operated in different regions[1] across United States of America (USA), it implies we need to validate safety of ML models in all these regions. Ensuring safety across different regions requires thorough representative data sets of all regions and corresponding ODD elements. Moreover, we must perform a

[1] A region can be a city or a county or a state or a group of states

Madala, K. and Do, H.
The Need for Location-based Machine Learning Models for Level 5 Automated Vehicles.
DOI: 10.5220/0010483706540661
In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2021)*, pages 654-661
ISBN: 978-989-758-513-5

thorough error analysis of these ML models and understand the impact of outputs from ML models on the behavior of the automated vehicle after sensor fusion.

After the deployment, if we find a situation in which one of the ML models being used in an automated vehicle results in a crash under some ODD settings, we will need to update the ML model or modify software to ensure the risk is reduced. If we update the ML model, then before deploying the updated ML model into the automated vehicle, we should perform change impact analysis (Kretsou et al., 2021) to comply with ISO 26262 and ISO 21448. This process ensures that the updated ML model is not compromising the FuSa or SOTIF of the automated vehicle.

Such an analysis, where we need to ensure safety of an ML model and automated vehicle over multiple locations/regions, can be highly time-consuming. If the fleet of vehicles is operating in a limited zone, the complexity of analysis may be reduced. However, if the fleet of automated vehicles are heterogeneous, i.e., there are different vehicle types that use different technologies, the safety analysis can be even more challenging. This complexity can add more difficulties with maintaining global models whose instances are deployed across a company's fleet of vehicles.

When we analyze safety of ML models in automated vehicles, we need to consider the temporal nature of data, i.e., how data can change with time, the amount of data being collected from each vehicle and time taken to scrutinize it, and the coexistence of vehicles which have no driving automation or partial driving automation (SAE level 3 or below). These factors add further complications when using instances of the same global models across the fleet of cars of a company because data changes with time and driving behaviors of vehicles with no or partial driving automation can vary with locations.

To overcome these limitations and to reduce the complexity, we propose the idea of location-based ML models for automated vehicles. The location-based ML models will limit the ODD and thereby reduce the time taken for change impact analysis and safety assessment. By using location-based models, safety assessments can be done per each model, thereby approving releases along with certifications per region. Safety assessment per region can also aid companies in limiting the recalls of vehicles to a particular region when an issue is found only within that region. We detail the engineering aspects we need to take into account when we use location-based ML models for automated vehicles to achieve level 5 of driving automation. Under the assumption that all regions have respective location-based models, note that using location-based models does not make an automated vehicle belong to SAE level 4. The automated vehicle will still belong to SAE level 5 as it can still operate anywhere without relying on human intervention.

The rest of the paper is organized as follows. Section2 discusses ISO 26262 and ISO 21448, and what aspects we need to consider for ML safety analysis according to these standards. Section 3 details the concept of change impact analysis and its application to ML models to comply with ISO 26262 and ISO 21448. Section 4 describes location-based ML models and how they aid the companies from both engineering and safety point of views. Section 5 provides insights and potential future directions and finally, we conclude in Section 6.

2 ISO STANDARDS AND ML MODELS

Automated vehicles should be compliant with two ISO safety standards: ISO 26262 (ISO, 2018) and ISO 21448 (ISO/PAS, 2020). ISO 26262 is the functional safety (FuSa) standard for automotive vehicles and ISO 21448 is the standard for safety of the intended functionality (SOTIF). FuSa aims at reducing risks to an acceptable level, which can occur due to malfunctions of hardware and software in a vehicle. SOTIF, on the other hand, aims at reducing unknown risks to a reasonable level and identifies gaps in requirements. SOTIF is performed under the assumption that the system follows its requirements as stated. The violation of the expected behavior from requirements due to hardware and software malfunctions fall under FuSa.

In the case of an ML model, as part of ISO 26262, we need to ensure the hardware on which the ML model is deployed and software which the ML model uses are not susceptible to malfunctions. ISO 26262 also requires the safety analysts to ensure a change in ML hardware, ML software, or an ML model does not compromise the functional safety of the system. The standard also requires to ensure the configurations of ML hardware, ML software, and an ML model are compatible even after updates.

On the other hand, ISO 21448 requires to analyze an ML model considering all possible scenarios that are possible in the ODD in which its predictions can result in safety issues. This includes ODD conditions under which an ML model cannot perform well, incorrect predictions by an ML model, and limitations of an ML model because of the data used, the ML training process followed, and an ML model's dependencies on other ML models if it is relying on their input to take a decision. ISO 21448 is done in conjunction with ISO 26262. Hence, an update in an ML model should be done through change impact analysis considering both the standards. As a result, an updated ML model to be compliant with ISO 21448 requires a thorough analysis with respect to its predictions, ODD conditions, as well as its impact on any dependent ML components.

Note that a given ODD can have many operating environments, and within each operating environment many scenarios are possible. Also note that, a scenario in ISO 21448 is defined as a combination of scenes, events, and actions with a defined set of goals and values including vehicle maneuvers. According to ISO 21448, we can define every scenario as a temporal sequence of situations that can occur within an operating environment. An ML model makes a decision based on each situation. For example, if we consider a sequence of frames from a camera, the entire sequence can be considered a scenario, each frame can be considered a situation. Hence, validation of ML models will need to be done at a situation-level to gain enough confidence for possible scenarios across the ODD. Thus, analyzing safety of an ML model even within a limited ODD can be a time-consuming task. For automated vehicles with level 5 of driving automation, which need to be able to travel anywhere without human feedback, performing SOTIF analysis to ensure compliance with ISO 21448 will require to analyze a large number of scenarios and corresponding possible situations for them. Also note that in order to have compliance with ISO 21448, we should also demonstrate that the automated vehicle has a very low possibility to face unknown hazardous scenarios by exploring scenarios in the real world or by using probabilistic modeling and functional decomposition (ISO/PAS, 2020). Hence, performing a change impact analysis when an ML model is updated can require a significant amount of effort.

3 CHANGE IMPACT ANALYSIS

In this section, we describe the change impact analysis and what should be considered to perform a change impact analysis when an ML model is updated/modified in autonomous vehicles.

Change impact analysis (Kretsou et al., 2021) is the process of identifying the effect of a change in a system, typically a change in software. To date, many researchers proposed various mechanisms for performing change impact analysis of requirements (Goknil et al., 2014), UML models (Briand et al., 2003), object-oriented programs (Ren et al., 2004), and aspected oriented programs (Zhao, 2002). For example, Zhao (Zhao, 2002) proposed a program slicing based technique for performing change impact analysis of aspect oriented programs. The major underlying goal of these change impact analysis approaches is to identify if the new change added to a system can affect the prior verified functionality. ISO 26262 also requires change impact analysis when a component or sub-system in a vehicle is modified (ISO, 2018). While there are some existing approaches on performing change impact analysis in automotive systems (Cui et al., 2018; Käßmeyer et al., 2015; Durisic et al., 2013), none of them have considered ML models.

The change impact analysis of ML models needs to identify the effect of the updated ML models on the FuSa and SOTIF of the automated vehicles. This implies that we need mechanisms to identify if the updated model might result in any new kinds of incorrect predictions that the previous version of the ML model did not, and if the updated model is compatible with ML software and ML hardware. To better understand the change impact analysis of an ML model, let us consider an example of a wrong-way detection system in an automated vehicle as shown in Figure 1. As the figure shows the wrong way detection system is an ML model which classifies whether a vehicle is going in a wrong way based on the predictions received from the traffic sign identification system and road marking identification system, which are the other set of ML models.

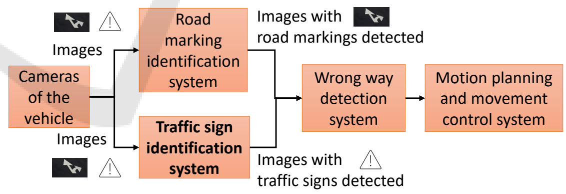


Figure 1: Example of a wrong way detection system of an automated vehicle, which uses inputs from traffic sign detection system and road marking detection system.

Let us assume that to detect a new sign, the traffic sign identification system's ML model is updated and deployed. In this case, as a part of change impact analysis, we should ensure the update in this ML model not only identifies the new sign but should also have acceptable performance for the other signs that are detected previously. We also need to ensure that the update in this model does not result in any incorrect predictions by a wrong way detection model, and that the updated model is compatible with the hardware and software on which it is deployed. We need to perform these steps to ensure the addition of the updated model does not lead to new hazards. If the updated model leads to new hazards, we should ensure they have a reasonable and acceptable level of risk. We can observe from these steps that the change impact analysis for an ML model even in a limited ODD can be a tedious and time-consuming process.

4 LOCATION-BASED MODELS

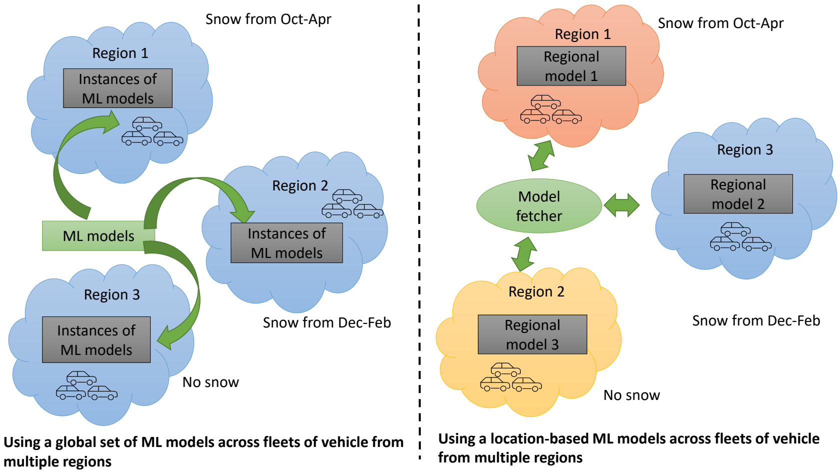
An autonomous vehicle can collect a large amount of data (in the order of terabytes) in one hour of operation. This data can also contain information related to inputs that resulted in crash sequences. Often such recently collected data including the crash sequence data are analyzed to propose the solutions to mitigate the crashes occurred. Such solutions can include retraining ML models by adding new data, using other sensor modalities in the car to mitigate the issues, and modifying the algorithms (e.g., sensor fusion related information). Hence, updating ML models can be commonly performed to reduce crashes or to improve performance and efficiency (Yang et al., 2018; Baylor et al., 2017).

Whenever an ML model is updated, as mentioned in the previous section, a change impact analysis should be performed to ensure that the changes in the ML model do not compromise safety. For safety assessment teams, in order to certify an autonomous vehicle, they need sufficient evidence, which can demonstrate that the updated model will not result in new safety issues that did not exist before. If we assume the fleet of vehicles to be operating in multiple regions in a country, then before deploying the same model across all fleets of vehicles and updating the models in all vehicles simultaneously, we have to make sure that the updated model does not result in crashes in all regions. If the vehicle company decides to recall the vehicles or shift to autonomous operation under remote monitoring, the car owners will suffer a great deal of inconvenience due to the absence or limited availability of these vehicles during such a service period, thereby damaging the company's reputations. To reduce the effort for performing change impact analysis, and restriction of operation to specific regions in the case of accidents, we propose to use location-based models.

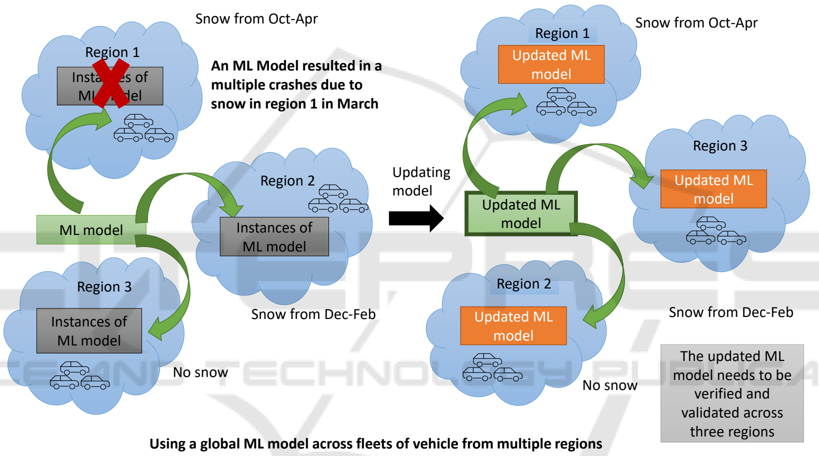
Figure 2 illustrates the approach that uses location-based models and how it is different from the approach that uses a global ML model across multiple regions. In this illustration, we considered three regions where fleets of vehicles are deployed. As described earlier, a global ML model approach (shown in Figure 2a) deploys the same model across different fleets in different regions. On the other hand, location-based models have region-specific ML models. If one of the vehicles is moving from one region to another, the vehicle should replace the previously used model with the new region's model. This will require a component that fetches the model relevant to the region (represented as "Model fetcher" in Figure 2a.

As shown in Figure 2, let us assume that fleets of vehicles are deployed across three regions such that region 1 has snowfall between October and April (e.g., St. Cloud in Minnesota), region 2 has no snowfall (e.g., Phoenix in Arizona), and region 3 has snowfall only between December and February (e.g., Baltimore in Maryland). Let us also assume multiple crashes were observed in March during snow in region 1 due to incorrect outputs from the ML model. In this case, for global ML model deployment as shown in Figure 2b, when we deploy the updated model after retraining into fleets of vehicles, it will need to be deployed across fleets in all three regions. As a result, we need to ensure the updated model will not compromise safety for any vehicle in the fleets of all three regions. This will require a comprehensive change impact analysis, simulation, verification, and sufficient evidence to show the new model will not increase the number of safety issues.

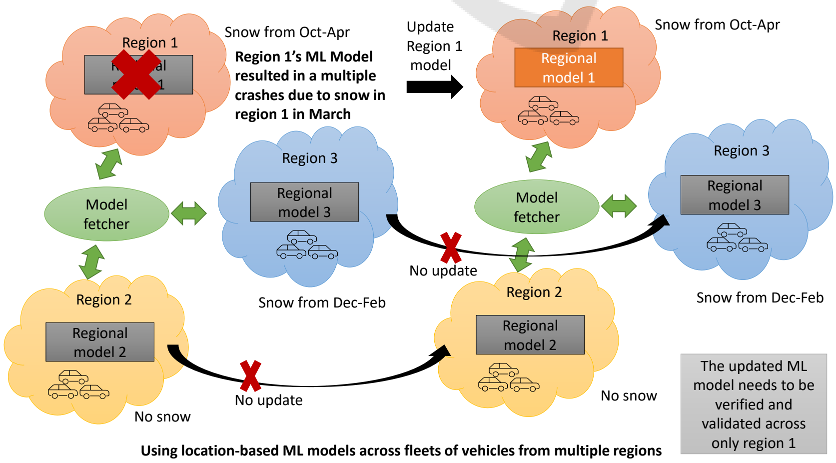
On the other hand, for location-based model deployment, if the regional model 1 in region 1 is resulting in safety issues due to snowfall in March, the other regional models need not be updated because 1) the regional model 1 is trained for ODD corresponding to region 1, and 2) regions 2 and 3 are not susceptible to snowfall in March. Hence, the fleets of vehicles in regions 2 and 3 can continue to operate without updating their ML models. Since we are only updating the ML model corresponding to region 1, we should sufficiently demonstrate safety by only considering region 1's ODD. The fleets in other regions do not need to wait for the updated model to be deployed across all regions to continue operation without compromising safety. This also helps in performing regional analyses of crashes, which makes it easier for analysts to perform error analysis and ablation studies of ML models. It also helps us to have a better understanding about how data changes in each region and thereby to consider modification at both local and global levels as needed. If such an understanding of how change of data with time needs to be analyzed at a global model level, it can be a very difficult task as currently only limited automation support is available to analyze such changes. Note that by using location-based model deployment, if a global model is intended to be released across all regions, it can be done region



(a) Global ML model deployment vs location-based ML model deployment.



(b) The effect of update in global ML model deployment.



(c) The effect of update in location-based ML model deployment.

Figure 2: Illustration of global ML model deployment and location-based ML model deployment, and the effect of updating a problematic ML model in both the approaches.

by region (e.g., the updated model can be released in region 1 initially and released eventually in regions 2 and 3 which do not have possibility for snowfall during March), thereby making the global model deployment convert to location-based model deployment.

Preliminary Analysis: To examine whether it is possible to have conditions that can be specific to a certain region, we conducted a preliminary analysis by considering 50 images picked randomly from nuScenes dataset (Caesar et al., 2020), and 50 images from Berkeley Deep Drive (BDD) dataset (Yu et al., 2020). The nuScenes dataset contains images collected from the front and rear cameras in Singapore and Boston, where as the BDD dataset has images collected from the front camera in New York, San Francisco, Berkeley, and Bay area. We used Yolo v4 (Bochkovskiy et al., 2020; Wang et al., 2020) pretrained model on the 100 images with the focus on identifying pedestrians, trucks, cars, and traffic lights. In nuScenes data, we found that for the image shown in Figure 3, pedestrians were not detected due to low visibility of people walking on a sheltered sidewalk. From BDD dataset, we did not find any images with a sheltered sidewalk similar to the one we found in nuScenes dataset. This implies that the presence of the sheltered sidewalk is found in Singapore or Boston, but not in other locations from which BDD dataset is collected. From this observation, we can infer that there can be ODD conditions that are specific to the regions. While we analyzed very few images, we believe that analyzing multiple datasets and model predictions from them might uncover ODD conditions that are available in one dataset but not other dataset, thereby indicating that location-based models can be beneficial. If we update the Yolov4 model to detect the pedestrians under sheltered pathways, we should reanalyze the conditions under which Yolov4 is unable to predict objects correctly. If such an updated model is used in an automated vehicle, we need to demonstrate if the updated model is going to result in any new hazards or not. If the error analysis of the updated model shows new patterns among incorrect predictions, we will need to verify the ML model for various possible scenarios that can occur within the ODD.

5 DISCUSSION

From the preliminary analysis, we can observe that location-based models can aid in moving towards SAE level 5 of driving automation. Both global ML model and location-based ML model deployment approaches have their own pros and cons as shown in Table 1. For diverse ODD settings, heterogeneous fleets of vehicles, change of data with time, consideration of behavior of other road vehicles with no automation and partial automation, and V2X communications, it would be better to use a location-based ML models as it can reduce the effort and time for analysis when updates are performed to ML models in automated vehicles. It would also help in performing a thorough error analysis and in gaining a sufficient confidence that the automated vehicle is safe enough to be certified.

Open Questions and Future Directions: While location-based ML model deployment is helpful, there are also some open questions and directions we should investigate.

1. *Mechanisms to Change the ML Model with Regions:* When using location-based deployment, if one vehicle currently operating on one region needs to travel to another region, the vehicle will need to replace its ML model once it reaches another region. However, when should the vehicle request for the model from the other region? What if there is no service available to remotely download the model at the entrance to another region? We need techniques and look-a-head scheduling strategies to figure out when we need to replace the model from previous region with the model from the region we are entering without affecting the continuous flow of driving. Should we use combination of the models during the transition from one region to another, or should models be trained by having some common boundary space than a strict line of separation between regions? We should investigate to find answers to these questions. We also need to consider means to ensure a vehicle's safety if the current model in the region is not available or redacted due to some compliance reasons. In such a case, we need to decide if we continue to use the previous model or shift to an emergency global safety model, which operates with a vehicle traveling by choosing lower speed roads and areas.
2. *Mechanisms to Ensure Compatibility of ML Models Across Regions with ML Hardware and Software:* When an automated vehicle enters a new region and the ML model from the region is downloaded to be used, the updated ML model should have compatibility with ML hardware and software in the car. If we are not able to ensure this function automatically, the safety of the automated vehicle can be compromised. Hence, research efforts in this direction should be considered.
3. *Granularity of Locations:* One of the open-end

(a) Original image with pedestrians under sheltered sidewalk highlighted.

(b) Predictions after applying the pre-trained Yolo v4 model to the original image.

Figure 3: Example of image where pedestrians are not detected due to a location-based ODD condition.

Table 1: Pros and cons of global ML model deployment and location-based ML model deployment.

	Global ML model deployment	Location-based ML model deployment
Pros	• No need to replace the ML model as the vehicle moves from one region to another. • Once deployed the ML model can be used anywhere within the ODD. • Global model might be able to adapt to new ODD elements in a region, if they already exist in other regions.	• A problematic ML model within a location/region can be updated without affecting the operation of fleets in other regions/locations and without updating ML models in other regions. • It reduces the effort and time required for change impact analyses and paves the way for a thorough error analysis and gaining sufficient evidence for safety certification.
Cons	• Significantly a large amount of effort required for change impact analysis, error analysis, and data selection. • A failure of an ML model at one region if resulted in halting of vehicles may result in decreased reliability for the use of a company's service by users.	• Requires a mechanism to replace an ML model based on locations/regions. • Lessons learned from error analysis of one region may not be transferred to other regions.

questions for location-based models is "what level of granularity should we have for a location or a region in location-based models?". Should we consider a location to be an area within a zipcode or a city/town/village or a county or a state or a group of states. We believe that having models based on the zones in the large cities and having models for the entire place for the small city, town, or village will be a good starting point. However, more research and studies should be conducted to identify the suitable and beneficial level of granularity.

6 CONCLUSIONS

In this paper, we proposed and discussed the location-based ML model approach in order to achieve safe deployment for automated vehicles with SAE level 5 of driving automation. We discussed its pros and cons compared to deploying the same set of ML models across fleets of vehicles operating in different locations.

As discussed in the previous section, there are many research directions with this topic. As part of future work, we plan to conduct a thorough empirical study on how location-based model and global ML model deployment processes affect the change impact analysis, system engineering process, and availability of vehicles. We also plan to explore what factors we need to consider for self-learning ML models to assure safety of the vehicle.

ACKNOWLEDGEMENTS

We thank Carlos Avalos-Gonzalez from kVA by UL for feedback on the topic.

REFERENCES

Aradi, S. (2020). Survey of deep reinforcement learning for motion planning of autonomous vehicles. *arXiv preprint arXiv:2001.11231*.

Baylor, D., Breck, E., Cheng, H.-T., Fiedel, N., Foo, C. Y., Haque, Z., Haykal, S., Ispir, M., Jain, V., Koc, L., Koo, C. Y., Lew, L., Mewald, C., Modi, A. N., Polyzotis, N., Ramesh, S., Roy, S., Whang, S. E., Wicke, M., Wilkiewicz, J., Zhang, X., and Zinkevich, M. (2017). Tfx: A tensorflow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1387–1395, New York, NY, USA. Association for Computing Machinery.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Briand, L. C., Labiche, Y., and O'Sullivan, L. (2003). Impact analysis and change management of uml models. In *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings.*, pages 256–265.

BSI/PAS (2020). 1883:2020. *Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification*.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631.

Chen, C., Seff, A., Kornhauser, A., and Xiao, J. (2015). Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730.

Committee, S. O.-R. A. V. S. et al. (2018). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE Standard J*, 3016.

Cui, L., Hu, J., Park, B. B., and Bujanovic, P. (2018). Development of a simulation platform for safety impact analysis considering vehicle dynamics, sensor errors, and communication latencies: Assessing cooperative adaptive cruise control under cyber attack. *Transportation Research Part C: Emerging Technologies*, 97:1–22.

Durisic, D., Nilsson, M., Staron, M., and Hansson, J. (2013). Measuring the impact of changes to the complexity and coupling properties of automotive software systems. *Journal of Systems and Software*, 86(5):1275–1293.

Goknil, A., Kurtev, I., Van Den Berg, K., and Spijkerman, W. (2014). Change impact analysis for requirements: A metamodeling approach. *Information and Software Technology*, 56(8):950–972.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386.

ISO (2018). 26262. *Road Vehicles – Functional Safety*.

ISO/PAS (2020). 21448. *Road vehicles - Safety of the intended functionality*.

Käßmeyer, M., Schulze, M., and Schurius, M. (2015). A process to support a systematic change impact analysis of variability and safety in automotive functions. In *Proceedings of the 19th International Conference on Software Product Line*, SPLC '15, page 235–244, New York, NY, USA. Association for Computing Machinery.

Kretsou, M., Arvanitou, E.-M., Ampatzoglou, A., Deligiannis, I., and Gerogiannis, V. C. (2021). Change impact analysis: A systematic mapping study. *Journal of Systems and Software*, 174:110892.

Pfeiffer, M., Schaeuble, M., Nieto, J., Siegwart, R., and Cadena, C. (2017). From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1527–1533.

Ren, X., Shah, F., Tip, F., Ryder, B. G., and Chesley, O. (2004). Chianti: A tool for change impact analysis of java programs. *SIGPLAN Not.*, 39(10):432–448.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2020). Scaled-yolov4: Scaling cross stage partial network. *arXiv preprint arXiv:2011.08036*.

Yang, K., Wang, X., and Yu, R. (2018). A bayesian dynamic updating approach for urban expressway realtime crash risk evaluation. *Transportation Research Part C: Emerging Technologies*, 96:192–207.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645.

Zhao, J. (2002). Change impact analysis for aspect-oriented software evolution. In *Proceedings of the International Workshop on Principles of Software Evolution*, IWPSE '02, page 108–112, New York, NY, USA. Association for Computing Machinery.