KnowGo: An Adaptive Learning-Based Multi-model Framework for Dynamic Automotive Risk Assessment

Paul Mundt[1], Indika Kumara[2,3](✉), Willem-Jan Van Den Heuvel[2,3], Damian Andrew Tamburri[2,4], and Andreas S. Andreou[5]

[1] Adaptant Labs, Adaptant Solutions AG, Berlin, Germany
paul.mundt@adaptant.io
[2] Jheronimus Academy of Data Science, Sint Janssingel 92, 5211 DA 's-Hertogenbosch, The Netherlands
[3] Tilburg University, Warandelaan 2, 5037 AB Tilburg, The Netherlands
{i.p.k.weerasinghadewage,w.j.a.m.vdnHeuvel}@tilburguniversity.edu
[4] Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands
d.a.tamburri@tue.nl
[5] Cyprus University of Technology, 3036 Limassol, Cyprus
andreas.andreou@cut.ac.cy

Abstract. In autonomous driving systems, the level of monitoring and control expected from the vehicle and the driver change in accordance with the level of automation, creating a dynamic risk environment where risks change according to the level of automation. Moreover, the input data and their essential features for a given risk model can also be inconsistent, heterogeneous, and volatile. Therefore, risk assessment systems must adapt to changes in the automation level and input data content to ensure that both the risk criteria and weighting reflect the actual system state, which can change at any time. This paper introduces *KnowGo*, a learning-based dynamic risk assessment framework that provides a risk prediction architecture that can be dynamically reconfigured in terms of risk criterion, risk model selection, and weighting in response to dynamic changes in the operational environment. We validated the *KnowGo* framework with five types of risk scoring models implemented using data-driven and rule-based methods.

Keywords: Dynamic risk assessment · Adaptive systems · Autonomous vehicles · Meta-learning · Multi-model · Dynamic software architecture

1 Introduction

Emerging data-driven business services in the insurance and transportation industries, such as usage-based insurance and risk-based pricing, require the

European Commission grant no. 825480 (H2020), SODALITE and no. 857420 (H2020), DESTINI.

B. Shishkov (Ed.): BMSD 2022, LNBIP 453, pp. 268–278, 2022.
https://doi.org/10.1007/978-3-031-11510-3_18

accurate assessment of the risk exhibited by a driver or vehicle on a given journey or overtime. For example, fleet managers can use the risk score as an indicator of driver safety or as a basis for driver coaching. Likewise, insurance companies can use it to calculate premium changes over time more accurately, allowing them to target discounts at careful or more experienced drivers and penalties at more aggressive ones.

Assessing automotive risk in real-time is problematic due to several complexities [2,15]. First, the same risk factor can have different implications depending on the level of automation and the driving situation. For example, harsh braking when a driver is fully in control may be assessed as high risk as it could result from distracted or defensive driving, each having different implications for the driving behavior while posing a similar level of situational risk to the vehicle. When operating at higher levels of automation in which the vehicle is either fully or partially in control, the same risk may be evaluated at a much lower level, with the expectation that any such occurrences are purely defensive in nature. Second, there exist different levels of vehicle automation [6], and during the same journey, the automation level of a vehicle can change multiple times. Thus, the risk assessment needs to adapt to the changes in the automation level [2].

Recently, learning-based approaches have been developed to support dynamic risk assessment (DRA), where risk-relevant metrics of the current driving situation are monitored and used as input for risk prediction/scoring models [2,5,13,14]. However, the proposed approaches exhibit several limitations. First, they use a single, static machine learning model and assume that data points are consistently obtainable, measurable, and of similar granularity. Second, they do not consider the changes to the level of automation during a journey. As a result, a static risk scoring model is often insufficient to produce accurate results over a more extended period. Furthermore, different data sources capture different perspectives of the risk assessment, and thus the representative risk predictors can be produced when each data source is used separately.

In this paper, to address the above-mentioned limitations of the existing studies, we present the *KnowGo Score* framework, which supports dynamic selection, tuning, and fusion of multiple risk scoring models. *KnowGo Score* allows using multiple risk scoring models, each using different data sources and risk assessment criteria. At runtime, *KnowGo Score* can monitor the current driving situation (per vehicle), select the most appropriate set of risk scoring models for a given automation level and availability of data, and combine the predictions made by the selected models to generate a weighted risk assessment. The implementation of the framework is available as a partially open-source product[1]. We assessed the practicability and usefulness of the framework by using a set of risk predictors and a vehicle/driving simulator.

This paper is structured as follows: Sect. 2 summarizes state of the art in the data-driven automotive DRA while highlighting their research limitations. Section 3 presents our *KnowGo Score* DRA framework in detail. Finally, Sect. 4 evaluates our framework, and Sect. 5 concludes the paper while outlining future research directions.

[1] https://knowgo.io/products/knowgo-score/.

2 Related Work

The data-driven approaches have been applied to predict automotive risk levels at runtime [2,13,15,18]. Feth *et al.* employed a Convolutional Neural Network (CNN) to predict the risk level of a driving situation based on the camera images of that particular situation [2]. They created the data set for training CNN models by simulating different driving situations with a driving simulator and assigning a risk metric to those situations. Then, they modeled the risk prediction as a regression problem. Kato [18] also used a CNN model but modeled the risk prediction as a classification problem. In [13], the Support Vector Machine (SVM) was used to predict the severity and controllability rating classes based on the measured data from the sensors. SINADRA [15] provides a framework for creating situation-aware dynamic risk assessment monitors. It uses a Bayesian Network Model for inferring a risk index from monitored risk factors. Katrakazas *et al.* [7] proposed a data-driven method that can estimate collision risk by considering road network safety information and inter-vehicle dependencies. They first used the ML classifiers to predict the network-level collision risk. They then calculated collision probabilities by integrating the predicted risk with inter-vehicle dependencies using dynamic Bayesian networks.

Liu*et al.* [10] used the real-time traffic features extracted from the Tweets to build probabilistic graphs that capture the causal relationships among the features and collision results. Next, a Bayesian network model based on those graphs is used to estimate the collision probabilities. Gao*et al.* [3] proposed a CNN model that can combine the information from both driving scene video data and kinematics data (*e.g.,* vehicle velocity and acceleration) to predict hazardous driving situations. Lin *et al.* [9] employed a set of classical ML and deep neural networks to predict accident risk locations (*e.g.,* intersections) using traffic accident data. The features in the data set include speed limit, road width, types of signs, pavement edge line, road patterns, and crossroads. They found road the first three features have the most impact on accidents.

The single static model approaches often assume that the input data are homogeneous, and the data points are consistently obtainable, measurable, and of similar granularity. However, this assumption does not always hold in practice. Machine learning models tuned to a given data set and features can, therefore, quickly become inadequate, which was also observed by some learning-based approaches for dynamic environments [17]. Another critical limitation of the learning-based risk prediction approaches is that they do not consider the changes in automation levels during a journey and the simultaneous co-existence of multiple vehicles at different automation levels.

3 KnowGo Automotive Risk Score Framework

To address the aforementioned limitations of the existing works, we propose the *KnowGo Score* framework, a system for assessing automotive risks at runtime. It implements a novel learning-based DRA architecture that enables dynamic

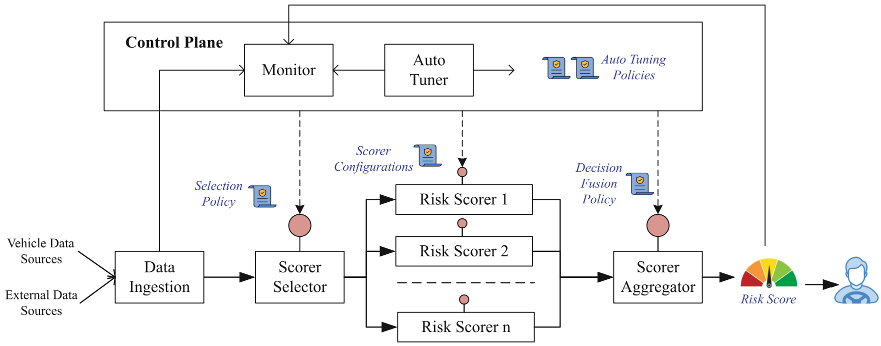

Fig. 1. Architecture of Dynamic Automotive Risk Assessment Framework

selection, tuning, and fusion of risk scoring models. Moreover, the framework allows multiple risk scorers to co-exist to support the heterogeneity in the risk assessment data sources. An accurate set of risk scoring models are dynamically selected based on the real-time data, and the decisions made by individual scorers are combined by applying decision fusion methods [11]. The rest of this section discusses the *KnowGo Score* framework in detail.

3.1 Overall Framework Architecture

Figure 1 depicts the architecture of the *KnowGo Score* framework, which consists of a set of *Risk Scorers* that can be dynamically enacted and managed. Each risk scorer calculates a risk index using one or more risk metrics based on the multi-dimensional input data from one or more data sources. The data can be injected into the system in real-time through the *Data Ingestion* component, which is a message-oriented middleware. *Scorer Selector* and *Control Plane* are the immediate consumers of the ingested data. The former component uses the contextual data extracted from the raw data,*e.g.,* automation level, and data source type, to choose the risk scorers for a set of data points. For this purpose, it uses a scorer selection policy that maps contextual attributes to risk scorers. The individual risk scorers predict risks as they receive the input data and send the predicted risk scores to the *Scorer Aggregator*, which in turn calculates the weighted average per automaton level, and produces the final averaged risk score. The decision fusion policy of the *Scorer Aggregator* defines risk scorers and scorer-specific weights suitable for assessing identified risks in the current vehicle state. At runtime, the *Monitor* at the control plane can observe the vehicle system state continuously and notify the *Auto Tuner* of state changes that impact risk assessment. These can include but are not limited to changes in automation level, legal jurisdiction, and driver privacy preferences. In response to state notifications, *Auto Tuner* can provide*Scorer Aggregator* and *Scorer Aggregator* with the updated policies for selecting scorers and fusing risk predictions. Each component of the *KnowGo Score* framework is implemented as a microservice that offers REST or event-driven APIs.

3.2 Risk Scoring Models

Individual risk scorers can be implemented using a range of data-driven techniques and rule-based techniques. As the scorers are not tightly coupled with the scoring framework itself, they are not limited by their choice of the implementation method. They may choose to use whichever technique is most appropriate for them. Section 4.1 describes the risk scorers currently available in the *KnowGo Score* framework. The *KnowGo Score* framework also supports adding and removing risk scorers at runtime via a plugin framework, where each scorer must implement the scorer interface defined by the framework.

The *KnowGo Score* system consists of general-purpose risk scorers and vehicle-specific scorers. The latter models can leverage additional vehicle manufacturer-provided data points. Risk Scorers are further broken down into three categories:*Independent Scorers*, *Dependent Scorers*, and *Augmented Scorers*. *Independent Scorers* are self-contained scoring models that can derive a risk score based on the input received, such as simple linear and logistic regression models. *Dependent Scorers* are scorers with finish-to-start dependency on one or more scorers in order to establish context for their risk assessment, if available. For example, a risk scorer can use driver alertness and obstacle detection to infer whether a harsh braking or swerving event is defensive or a result of driver inattentiveness. *Augmented Scorers* are scorers that extend the input data with external data to provide additional context for their risk assessment, such as using vehicle geolocation to obtain weather and road condition data.

3.3 Meta Risk Scoring Methods

Scorer Aggregator uses decision fusion methods to combine decisions made by individual risk scorers. The final fusion output (*i.e.,* the overall automotive risk) depends on the prediction accuracy of the risk scorers and the fusion algorithm. There exist various fusion algorithms, including averaging and voting methods, data-driven models (*e.g.,* classical machine learning and deep neural networks), and rule-based methods [11,16]. In this study, we selected the averaging method, which is simple, intuitive, and used for integrating the decisions made by regression models [12].

We considered three variations of the averaging scheme: simple averaging, weighted averaging, and confidence weighted averaging. In the simple averaging method, the average value of the predictions of individual risk scorers is calculated for each trigger interval. In the weighted averaging scheme, the prediction of each risk scorer is multiplied by the weight given for the scorer, and then their average is taken. Finally, the confidence weighted averaging scheme further normalizes the scorer-specific weights by multiplying the weight with the prediction confidence rate, ensuring that uncertain predictions are not given the same weighting in the final risk score calculation as more confident ones. The formula for the confidence weighted averaging is:

$$W = \frac{\sum_{i=1}^{n} b_i c_i w_i X_i}{\sum_{i=1}^{n} b_i c_i w_i} \tag{1}$$

```
rule "from_3_to_4"
 when
  $f1 : AutomationLevelChangedEvent(preLevel == 3, newLevel == 4)
 then
  metaScorerAPI.reconfigure(new String [] {"JourneyDuration:10:40",
                                            "HarshBraking:10:40", "NightDriving:10:40",
                                            "DriverAlertness:10:40", "WeatherConditions:10:40"});
end
```

Fig. 2. A Snippet of the Rules Used by Auto Tuner

where b_i is the scorer-specific bias given for an individual driver or vehicle, c_i the confidence of a given prediction by a specific scorer, w_i the scorer-specific weighting, and X_i the scorer-specific risk score.

Weighting enables end-users to express their preferences on risk scoring models and their contributions. For example, in the case where a vehicle is switched to the autonomous driving mode, the weighting of a driver monitoring risk scorer can be increased or decreased, proportional to the autonomous driving level, establishing a direct correlation between the degree of risk of inattentive or distracted behavior by the driver with the level of automation. This change in weighting similarly allows driver alerting to adapt and escalate/deescalate in severity. A driver or vehicle-specific bias may also be applied to increase or decrease the impact of individual scorers.

3.4 Auto-tuning

Auto Tuner decides and carries out the desired reconfigurations or tunings to the *Scorer Selector* and *Scorer Aggregator*. To allow the end-users to define the tuning decisions, we provide an ECA (Event-Condition-Action) policy language. A policy consists of a set of ECA rules. The rule-based systems are one of the most popular approaches to implementing self-adaptive systems [4]. The *Auto Tuner* continuously receives the vehicle state data, including the changes in the automation level and driver preferences, from the *Monitor* component. The reception of the vehicle data can trigger auto-tuning rules, which in turn enact the necessary changes to the behaviors of the *Scorer Selector* and *Scorer Aggregator* through their REST APIs. Currently, the changes are limited to the configurations of the decision fusion methods used by them, *i.e.,* changing the selection of risk scorers and their weights and confidence levels. Figure 2 shows an example of a rule that reacts to the event *AutomationLevelChangedEvent* by adjusting the weights and confidence levels of the individual risk scorers.

4 Implementation and Evaluation

We set the following research questions for evaluating *KnowGo Score*:

RQ1 - To what extent can risk scorers predict the automotive risk accurately?
RQ2 - To what extent can auto-tuning of risk scoring help to accurately predict risk as automation level changes?

Table 1. General-purpose Risk Scorers in the KnowGo Score System

Risk Scorer (Default Weight)	Description	Data Points Used
Night Driving (50)	Determine the extent of a journey that has taken place in night-time conditions	Sunset/Sunrise times at location GPS location and timestamp
Journey Duration (50)	Determine the duration of a journey, and calculate its risk relative to Regulation (EC) No 561/2006	Journey start/stop times, journey duration (for in-progress journeys)
Weather Conditions (65)	Determine the weather conditions during a journey and assess whether a vehicle is appropriately configured for the conditions	GPS location and timestamp, Vehicle make/model, Weather conditions, Instrumentation status
Driver Alertness (75)	Determine if a driver is drowsy or distracted. Includes heart rate monitoring, eyelid closure, and gaze estimation	In-cabin video, Heart rate from wearable sensor, gyroscope readings
Harsh Braking/ Acceleration (50)	Determine the extent to which harsh braking and acceleration events have occurred during the course of a journey	Accelerator/brake pedal positions and timestamps, gyroscope and accelerometer readings

4.1 Framework Implementation

We developed the *KnowGo Score* framework[2] using Python. To implement the risk scoring microservices, we used *Flask* micro web framework, *scikit-learn* machine learning library, and *OpenCV* computer vision library. We used the Drools business rule management system to implement the rule-based auto-tuning engine. All components are containerized with *Docker*. The framework can be deployed and managed on a *Kubernetes* cluster over Edge and/or Cloud infrastructures. To simplify the deployment and runtime adaptation processes, we used the SODALITE framework [1,8].

Table 1 and Table 2 provide a brief overview of the risk scorers implemented, including their data sources and their mappings to SAE(Society of Automotive Engineers) automation levels. As appropriate, we use both ML-based and non-ML algorithms. The domain experts at the KnowGo company decided on the risk calculation algorithm for each risk scorer. As regards ML algorithms, we selected a wide range of learning algorithms used by the research literature [2,9]: Linear Regression (Journey Duration), Logistic Regression (Harsh Braking/Acceleration), and Convolutional Neural Network (CNN) (Driver Alertness). The non-ML scorers include the Night Driving scorer and Weather Conditions scorer.

[2] https://knowgo.io/products/knowgo-score/.

Table 2. Mapping of SAE Automation Levels to Risk Scorers and Weights

Level	Risk Scorer (Default Adjusted Weight)	Justification
0	Driver Alertness (75), Journey Duration (50) Harsh Braking/ Acceleration (50)	No driver support, defaults unchanged
1	Driver Alertness (75), Journey Duration (50) Harsh Braking/ Acceleration (40)	Possible acceleration/braking or steering assistance to the driver
2	Driver Alertness (75), Journey Duration (50) Harsh Braking/ Acceleration (30)	Acceleration/braking and steering assistance provided by vehicle
3	Driver Alertness (85), Journey Duration (25) Harsh Braking/ Acceleration (15)	Vehicle takes over driving tasks, driver must maintain situational awareness, risk of automation complacency
4	Driver Alertness (5), Journey Duration (15) Harsh Braking/ Acceleration (15)	Vehicle maintains situational awareness, driver involvement significantly reduced
5	Harsh Braking/Acceleration (15), Journey Duration (15)	No driver involvement, fully autonomous

4.2 Dataset Generation

As it is challenging to collect a large amount of data for multi-user and multi-autonomy scenarios in a real driving environment, as in [2,13], we used a simulation environment to generate sufficient data for training and evaluating risk score prediction models. A range of journeys, events, and instrumentation readings under several different driving conditions and styles have been generated with the *KnowGo Vehicle Simulator*[3], an open-source connected car simulator designed to generate realistic streaming vehicle telemetry. The simulator can also track and expose changes to the level of automation to enable the generation of journeys with events spanning across multiple levels of automation.

4.3 RQ1: Accuracy of Risk Scorers

Table 3 shows the average overall accuracy of the scoring models, including *Scorer Aggregator*. The accuracy of the *Scorer Aggregator* is directly influenced by its included scoring models, and is brought up or down with the introduction of additional models. As both the automation level and the overall model confidence directly impact the per-model weighting, lower confidence results in lower overall impact on the final risk score, ensuring that the impact of reduced accuracy for a given model can be better absorbed by the system without having a significant

[3] https://github.com/knowgoio/knowgo-vehicle-simulator.

Table 3. Predictive Accuracy of the Risk Scorers

Risk Scorer	Model	Accuracy
Journey Duration	Linear Regression	72.05
Harsh Braking/Acceleration	Logistic Regression	95
Night Driving	Non-ML	100
Weather Conditions	Non-ML	100
Driver Alertness	Multi-model, Linear SVM, CNN	75
Scorer Aggregator	**Performance-weighted-voting**	**93.75**

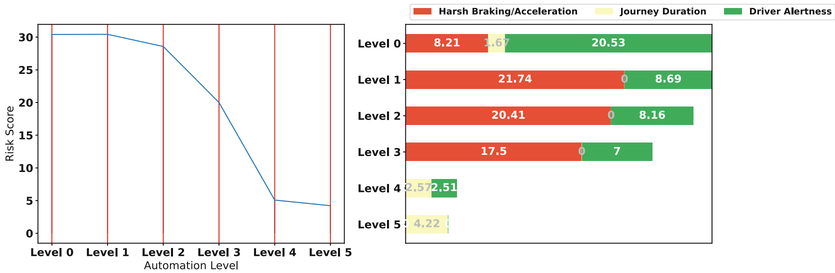

Fig. 3. (a) Risk Score vs Automation Level, (b) Scorer Contribution

impact on the underlying score. A mixture of models including ML and non-ML ones exhibiting a high degree of accuracy further ensure that confidence variance can be handled while keeping the overall confidence high.

4.4 RQ2: Effectiveness of Auto-tuning of Risk Scorers

Figure 3 shows the changes in the risk score in response to changes in automation levels and the contribution of the risk scorers at each level. This experiment only used ML-based risk scorers. In (a), we observe that the risk level for a journey drops in proportion to the level of automation. Per the scorer selection and weighting outlined in the previous section, the overall risk drops off considerably as the driver's role is diminished. A notable exception is level 3, in which the driver must be alert and ready to intervene, and the risk of automation complacency emerges. In (b), we see that driver alertness has significantly more impact at level 3 than at higher levels. Journey duration risks are lessened on a per-level basis, as this reflects the journey duration risk for the time spent within a specific level of automation, which is later compounded. By the time the journey is long enough for this to become more of a risk, higher levels of automation have taken over, and the risk is mitigated.

5 Conclusion

This paper proposes, *KnowGo Score*, a novel dynamic automotive risk assessment framework based on runtime selection and fusion of risk prediction models. It supports assessing the automotive risk levels in a dynamic environment, where automation level and input data frequently change over time. To improve the accuracy of the overall automotive risk assessment, the framework employs multiple risk scoring models that use different data sources and learning algorithms. A scheme for selecting and ranking scoring models according to changing data points and automation levels is presented. A partially open-source implementation of the *KnowGo Score* is available. With a set of risk predictors and a vehicle simulator, the practicability and usefulness of the framework were assessed. We plan to extend our risk assessment framework for future work by supporting different meta-learning approaches and incorporating more adaptation capabilities such as switching off/on or scaling up/down models on demand. We will also investigate monitoring and adapting to complex situations or context changes.

References

1. Di Nitto, E., et al.: An approach to support automated deployment of applications on heterogeneous Cloud-HPC infrastructures. In: 2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 133–140 (2020)
2. Feth, P.: Dynamic Behavior Risk Assessment for Autonomous Systems. Ph.D. thesis, Kaiserslautern University of Technology, Germany (2020)
3. Gao, Z., Ou, M., Liu, Y., Zheng, J.Y.: Perceiving driving hazards in a data-fusion way using multi-modal net and semantic driving trajectory. In: 2020 International Conference on Sensing, Diagnostics, Prognostics, and Control, pp. 322–328 (2020)
4. Ghahremani, S., Giese, H., Vogel, T.: Efficient utility-driven self-healing employing adaptation rules for large dynamic architectures. In: 2017 IEEE International Conference on Autonomic Computing (ICAC), pp. 59–68 (2017)
5. Hegde, J., Rokseth, B.: Applications of machine learning methods for engineering risk assessment - a review. Saf. Sci. **122**, 104492 (2020)
6. SAE International: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. SAE (2018)
7. Katrakazas, C., Quddus, M., Chen, W.H.: A new integrated collision risk assessment methodology for autonomous vehicles. Accid. Anal. Prev. **127**, 61–79 (2019)
8. Kumara, I., et al.: SODALITE@RT: orchestrating applications on cloud-edge infrastructures. J. Grid Comput. **19**(3), 29 (2021). https://doi.org/10.1007/s10723-021-09572-0
9. Lin, D.J., Chen, M.Y., Chiang, H.S., Sharma, P.K.: Intelligent traffic accident prediction model for internet of vehicles with deep learning approach. IEEE Trans. Intell. Transp. Syst. 1–10 (2021)
10. Liu, X., Lan, Y., Zhou, Y., Shen, C., Guan, X.: A real-time explainable traffic collision inference framework based on probabilistic graph theory. Knowl.-Based Syst. **212**, 106442 (2021)
11. Mangai, U.G., Samanta, S., Das, S., Chowdhury, P.R.: A survey of decision fusion and feature fusion strategies for pattern classification. IETE Tech. Rev. **27**(4), 293–307 (2010)

12. Mendes-Moreira, J., Soares, C., Jorge, A.M., Sousa, J.F.D.: Ensemble approaches for regression: a survey. ACM Comput. Surv. **45**(1), 1–40 (2012)
13. Patel, A., Liggesmeyer, P.: Machine learning based dynamic risk assessment for autonomous vehicles. In: International Symposium on Connected and Autonomous Vehicles (SoCAV) (2021)
14. Rabe, M., Milz, S., Mader, P.: Development methodologies for safety critical machine learning applications in the automotive domain: a survey. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 129–141 (2021)
15. Reich, J., Trapp, M.: SINADRA: towards a framework for assurable situation-aware dynamic risk assessment of autonomous vehicles. In: 2020 16th European Dependable Computing Conference (EDCC), pp. 47–50 (2020)
16. Sinha, A., Chen, H., Danu, D., Kirubarajan, T., Farooq, M.: Estimation and decision fusion: a survey. Neurocomputing **71**(13), 2650–2656 (2008)
17. Stefana, E., Paltrinieri, N.: Prometaus: a proactive meta-learning uncertainty-based framework to select models for dynamic risk management. Saf. Sci. **138**, 105238 (2021)
18. Wang, Y., Kato, J.: Collision risk rating of traffic scene from dashboard cameras. In: 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–6 (2017)