Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
17/1 (2024), 59-66. DOI: http://dx.doi.org/10.21609/jiki.v17i1.1199

A Dynamic-Bayesian-Network-Based Approach to Predict Immediate Future Action of an Intelligent Agent

Rinta Kridalukmana[1], Dania Eridani[1], Risma Septiana[1]

[1] Department of Computer Engineering, Faculty of Engineering, Universitas Diponegoro
Jl. Prof. Soedharto, S.H., Semarang, 50275, Indonesia

Email:rintakrida, dania@ce.undip.ac.id, rismaseptiana@live.undip.ac.id

Abstract

Predicting immediate future actions taken by an intelligent agent is considered an essential problem in human-autonomy teaming (HAT) in many fields, such as industries and transportation, particularly to improve human comprehension of the agent as their non-human counterpart. Moreover, the results of such predictions can shorten the human response time to gain control back from their non-human counterpart when it is required. An example case of HAT that can benefit from the action predictor is partially automated driving with the autopilot agent as the intelligent agent. Hence, this research aims to develop an approach to predict the immediate future actions of an intelligent agent with partially automated driving as the experimental case. The proposed approach relies on a machine learning method called naive Bayes to develop an action classifier, and the Dynamic Bayesian Network (DBN) as the action predictor. The autonomous driving simulation software called Carla is used for the simulation. The results show that the proposed approach is applicable to predict an intelligent agent's three-second time-window for immediate future action.

Keywords: *human-autonomy teaming, intelligent agent, human-agent interaction*

1. Introduction

In the last decade, the interaction between humans and an intelligent agent (IA) has become more sophisticated as more authorities are assigned to IA providing it with more autonomy based on its decision-making process. In this regard, IA is considered a human counterpart rather than merely a tool to support operational tasks [1]. Such an interaction is widely known as human-autonomy teaming (HAT), and the autonomy agent refers to an intelligent agent assigned a high level of autonomy in a certain task [2].

Even though IA is highly autonomous, most HATs still require human involvement in the driving loop control [3]. Driving loop control is a mechanism to control car maneuver which can be done by either the human driver or an intelligent agent. This occurs due to IA's limited capabilities on its sensory tools to form its perception, and on its learning model to form its situational awareness (SA). IA's SA is used to determine any necessary actions in response to given situations [4–6]. However, without a proper tool for humans to know IA's immediate future actions, it will cost human response time to go back to the driving loop control to take necessary actions over the car when the predicted IA's actions go south [7–9]. Level four of driving automation, namely partially automated driving (also called collaborative driving in this paper), according to the Society of Automotive Engineers is considered an example of HAT that can benefit from the immediate future action predictor for its autopilot agent [10–12]. Hence, developing such an action predictor is challenging and highly required, particularly in the HAT context.

To answer the challenge mentioned above, this research aims to develop an approach to predict the immediate future actions of IA. From literatures, such approaches have been proposed to fulfill requirements from the industrial field [13–16]. However, lack of studies that implement such action predictors in a more dynamic and unpredictable environment such as a collaborative driving context. For instance, a maneuver classification model has been proposed by [17, 18] to check whether or not the correct maneuver is recognized before the vehicle attends the intended lane. Such a classification model is significant as the fundamental part of the action predictor. Moreover, [19] proposed a turning maneuver model to predict turn left, turn right, and going straight behaviors. However, this proposal is developed for connected vehicles environment. Hence, to fill the research gap, the collaborative driving context is selected in this study for the experimental case.

This research uses an autonomous driving simulator software called Carla for the experiment. As driving situations are very complex, this study selects an overtaking situation case to simulate the action predictor. Furthermore, the proposed approach involves two main techniques, the Naive Bayes and Dynamic Bayesian Network (DBN) for their capabilities to save computational cost and to support missing data, respectively. Hence, the combination of two techniques to predict the immediate future action of an intelligent agent is considered the main contribution of this study. The experiment results indicate that the proposed approach is applicable to predict immediate future actions.

The rest of this paper is structured as follows. Section 2 presents related works. Section 3 represents the proposed method. Section 4 provides the experiment and results. Lastly, conclusions and future works are drawn in Section 5.

2. Related Works

Most literature that discusses IA's action prediction is very close to studies about situation assessment, particularly in the industrial area. In this regard, hazardous and abnormal situations are identified [15], and IA calculates the consequences in the immediate future to execute safety procedures accordingly [16]. Similarly, [14] also viewed that future consequences must be included as a part of alarm design. However, such consequences must be prioritized by considering human cognitive abilities to absorb critical safety information generated by IA [13]. The action prediction approaches in the industrial field are quite mature as those approaches have been deployed under strictly controlled circumstances.

Besides the previously mentioned studies above, the IA's action identification is embedded in the topic of IA's self-explaining abilities. For example, the proposed approach from [6] is based on a provenance graph to explain IA's behaviors. The self-explaining ability tested in a collaborative driving context is also demonstrated in [20]. In this regard, IA is the autopilot agent (AA). However, these approaches are considered data-driven approaches, so they present historical data on AA. The development of an action predictor is inline with the human-vehicle collaboration framework proposed by [21, 22]. According to the frameworks, a mutual understanding module such as an action predictor, is required to ensure that the automation actions and maneuvers are predictable to humans and vice versa. Hence, developing a predictor for IA's immediate future action needs to be explored.

3. Proposed Method

This section provides a detailed process to develop a machine-learning-based classifier to predict AA's immediate future actions in the one-way street overtaking scenario. This section is divided into three parts. The first part explains how data is generated and prepared for the training dataset. The second part presents the way to develop an action predictor. Finally the final part provides the way the action predictor is evaluated.

3.1. Dataset Generation and Preparation

This research used an open-source autonomous car simulation software called Carla (version 9.13) to generate the dataset. The attributes of the dataset follow the research from [20], that can be seen in Table 1. Those attributes obtain their values from code-generated flags during the autonomous car's overtaking task. For example, the attributes of ρ and υ are flags from the lane change warning system. The flag for driver approval (ψ) is assumed when the AA starts its overtaking maneuver. Furthermore, the flag for overtaking speed risk (ζ) is obtained from the road speed limit provided by the navigation system. The other attributes are collected with the help of Carla's virtual sensors including LIDAR. As shown in Table 1, all flags are indexed based on the attribute's possible values.

Attributes in the dataset are considered the situation of concern's attributes. Several instances of Carla simulation are executed to obtain the values of those situation attributes. The value's combination of those attributes is then used to predict the immediate future action of the autonomous car's autopilot agent. The overtaking scenario simulated in Carla's environment is on a one-way road. For such a scenario, there are six typical actions of AA in association with overtaking tasks (see Table 2): 1) keep going. 2) overtaking aborted but stay

Table 1. Attributes of the dataset and their description.

Attribute Symbols	Description	Possible values and their indices
ρ	Position of ego vehicle relative to overtaking car	Behind [0], Next To [1], After [2]
υ	Current lane position of the ego vehicle	Overtaking lane [0], Ego lane [1], Departure Lane [2]
ψ	Driver's approval on the autopilot overtaking task	Yes [0], No [1]
ζ	Overtaking risk against road speed limit	Safe [0], Unsafe [1]
η	The existence of other cars in the overtaking lane	Exist [0], Not Exist [1]
ϑ	Distance or collision risk against η	Safe [0], Unsafe [1]
ε	Risk related to space to go back to the departure lane after overtaking	Safe [0], Unsafe [1]

Table 2. Labels and Symbols used to label the autopilot agent's actions.

Label	Symbol	Description
0	KG	Keep going with the current action
1	C1	Overtaking abort, stay in overtaking lane
2	C2	Overtaking abort, back to ego lane
3	KP	Keep processing the overtaking task
4	LC	Doing lane change
5	GT	Go back to the departure lane

Table 3. The snippet example of the dataset to classify the autopilot agent's action.

ρ	υ	ψ	ζ	η	ϑ	ε	label
2	0	0	0	0	0	0	5
1	1	0	1	0	0	0	4
1	0	0	0	1	0	0	3
1	2	1	0	1	0	0	0
0	0	0	1	1	0	0	1
2	0	0	1	1	0	0	2

in the overtaking lane, 3) overtaking aborted and go back to departure lane, 4) keep processing the overtaking task, 5) lane changing, and 6) go to departure lane. Based on the typical actions, each raw data generated by Carla instances are labeled. The snippet of the generated dataset can be seen in Table 3. The example overtaking scene in the Carla simulator to generate a dataset can be seen at https://youtu.be/gU4vwh6bSyg. Another example when overtaking risk against the road speed limit as the overtaken vehicle increases its speed can be seen in video link: https://youtu.be/VbqYyaKZEpA.

3.2. Action Predictor

The action predictor is developed using two fundamental techniques, namely machine learning and DBN. The working scheme of the proposed action predictor is illustrated in Fig. 1. The previously generated dataset will be trained using a machine learning technique to generate a classifier model for AA's actions. Among many machine learning approaches, this research selects naive Bayes to train the model because of its feature in reducing the computational cost. The results of the classifier generated by the naive Bayes will be used to feed the DBN. DBN is exploited to predict the immediate future value of the AA's actions. DBN is a well-known technique for data relationships against time sequences. Moreover, it also supports missing data. Hence, DBN is suitable to address sensor missing data problems which often occur while IA collects the data from its sensory tools. The fundamental of each technique used in this study is described below.

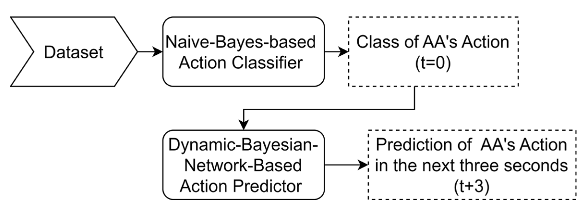

Figure 1. Working scheme of the proposed method.

3.2.1. Naive Bayes. Naive Bayes is one of the supervised learning techniques that apply the Bayes theorem. Naive Bayes uses a 'naive' assumption in which between every tuple of feature given random variable's values, there would be a conditional independence so that:

$$P(a_i|b, a_i, \ldots, a_{i-1}, a_{i+1}, \ldots, x_n) = P(a_i|b) \quad (1)$$

where b is a random variable, and a_1 through a_n are feature vector that has dependencies on each other. As $P(a_1, \ldots, a_n)$ is constant, the following classification rule can be applied:

$$\hbar = \arg\max_{b} P(b) \prod_{i=1}^{n} P(Pa_i|b) \tag{2}$$

where $P(Pa_i|b)$ is considered the recurrence rate of class b in the training dataset. Then, the Gaussian is implemented to calculate the likelihood of the feature as denoted by:

$$P(a_i|b) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(a_i - \mu_b)^2}{2\sigma_b^2}\right) \tag{3}$$

where σ_b and μ_b are based on maximum likelihood estimation.

In this research, the dataset is trained using the Naive Bayes. As illustrated in Fig. 1, the output of the Naive Bayes is a classifier describing the AA's actions (see Table 2).

3.2.2. Dynamic Bayesian Network. The Bayesian Network (BN) is considered a directed acyclic causal network that consists of nodes corresponding to random variables and arcs representing causal influences among variables. Joint probability distribution $P(V)$ is calculated among the BN's random variables $V = \{v_1, v_2, \ldots, v_n\}$, and it is formulated by:

$$P(V) = \prod_{i-1}^{n} P(v_i|Pa(v_i)) \tag{4}$$

where $Pa(v_i)$ denotes the parent of V_i for the finite number of $i = 1, 2, \ldots, n$. If the parent is empty, then:

$$P(v_i|Pa(v_i)) = P(v_i) \tag{5}$$

which represents its prior probability. Based on Bayes theorem, prior probability can be updated when new evidence E comes during the system's activities and operations, thus the posterior probability is calculated by

$$P(V|E) = \frac{P(V,E)}{P(E)} \tag{6}$$

DBN is an extension of the BN's capabilities, and it can capture dynamic changes within domain variables of the static BN at different times. However, DBN can distinguish variables within both the same period and the different periods. The former is called contemporaneous dependency, and the latter is referred to as non-contemporaneous dependency. Hence, a DBN can be defined as a tuple (B_1, B_T) where B_1 denotes a BN having $P(v_1)$ as the prior distribution and B_T indicates a two-slice temporal BN with so that:

$$P(V_t|V_t - 1) = \prod_{i-1}^{n} P(v_t^i|Pa(v_t^i)) \tag{7}$$

where v_t^i denotes the node state at time-window t and $Pa(v_t^i)$ denote the parent nodes in time-window t or $t-1$. Thus, the joint probability distribution can be formulated as:

$$P(v_1 : T) = \prod_{t=1}^{T} \prod_{i=1}^{n} P(v_t^i|Pa(v_t^i)) \tag{8}$$

where T represents the time slices in total.

In this research, DBN receives input from the Naive Bayes. The output of DBN is the prediction of AA's action in the next three seconds.

3.3. Performance Evaluation

There are two main evaluations in this research. The first evaluation is to measure the accuracy of the classification performance of the AA's action developed using the Naive Bayes. The second evaluation is to check the performance of the DBN to predict the immediate future action of AA. In this regard, this research defines immediate future action as a three-second time window in the future. The three-second unit is selected because it is considered to be the minimum time needed by a driver to comprehend a driving situation.

4. Experiment and Results

4.1. Action Classifier Performance

As previously mentioned, the dataset generated by Carla simulator will be trained to classify the category of AA's action as presented in Table 2. Table 4 presents the classification report that includes precision, recall, f1-score and accuracy measurements. For accuracy, the classifier achieves 98% valid predictions, and its confusion matrix can be seen in Fig. 2. It can be seen in Table 4 that mostly, the dataset contains *keep going* (KG, labeled by 0) actions which is normal as in most overtaking situations AA deals with normal situations. Also, the sensor captures the data every 0.25 seconds, hence KG gets a lot of support data. Other actions only have few support data because they only occurred in a very short time window during the simulation. Furthermore, this research also verifies the performance of the action classifier performance with k-fold cross-validation technique with the k value is

Table 4. Classification report on autopilot agent's action after the last k.

Class	Precision	Recall	f1-score	support
0	1.00	1.00	1.00	1494
1	1.00	0.79	0.89	131
2	0.93	0.91	0.92	210
3	0.67	1.00	0.80	39
4	1.00	1.00	1.00	401
5	0.71	1.00	0.83	29
accuracy			0.98	2304
macro avg	0.88	0.95	0.91	2304
weighted avg	0.98	0.98	0.98	2304

Table 5. K-Fold Cross Validation Score.

Folds	Mean Score	Min Score	Max Score
1	0.98	0.982	0.983
2	0.98	0.98	0.986
3	0.98	0.978	0.93
4	0.98	0.98	0.987
5	0.98	0.98	0.987

set to 5, and hence, the training and test dataset will be 80% and 20%, respectively.

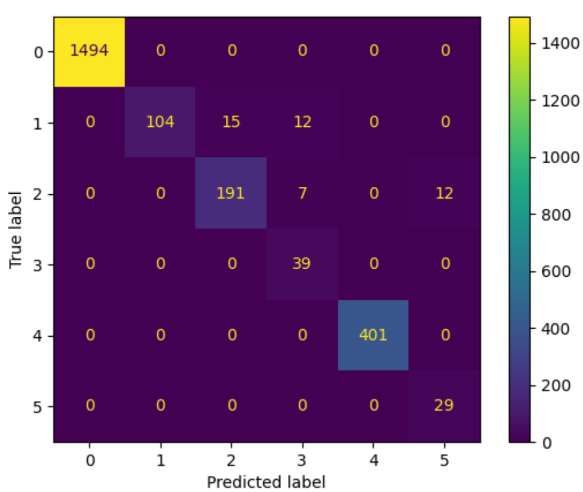

Figure 2. The confusion matrix of autopilot agent's action classification after the last k.

Among identified actions, $C1$ and $C2$ are identified, particularly when the AA deals with hazardous situations during its overtaking task, which is indicated by attributes ζ, η, and ε as presented in Table 1. In this research, those two types of actions are highlighted as they can help the drivers enhance their awareness and understanding of AA during overtaking task cancellation caused by hazardous situations.

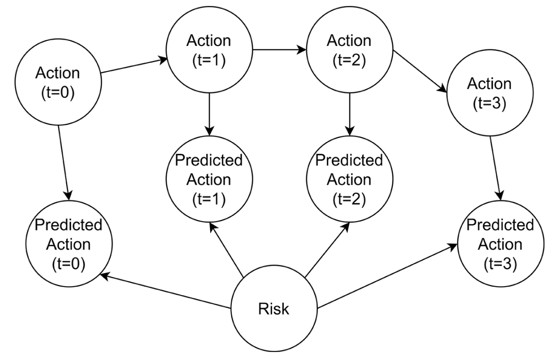

Figure 3. The Relations among nodes in the Dynamic Bayesian Network to predict AA's action in the next three seconds.

4.2. Action Predictor Performance

After AA's actions are conceptually labeled, this section presents the action predictor performance which takes advantage of the DBN. The performance is measured based on whether or not, in the next three seconds, the predicted action will have the same label as the real action. The action predictor engine is then developed based on the DBN, and the relations among nodes in the DBN used to predict AA's actions in the next three seconds are illustrated in Fig. 3. The DBN has two node types, namely temporal and static nodes. Furthermore, the DBN involves three nodes, namely the *Action* node, *Predited Action* node, and *Risk* node. The two former nodes are temporal nodes, and the latter is a static node.

The time step count (t) in the DBN model is set to four (4), from t=0 to t=3, to represent the next three seconds. The current action is considered an action at t=0, and its label will become the input for the *Action* node at t=0. The *Predicted Action* node at t=3 is the node representing what the AA's action will do in the next three seconds. For example, at t=0, the recognized action by the Naive-Bayes-based classifier is KP. However, the predicted action shown by the *Predicted Action* node could be $C1$, $C2$, or KP depending on the situation changes indicated by the *Risk* node. Furthermore, the arc order between the two temporal nodes, the *Action* node and the *Predicted Action* node, is set to 1 to indicate that the time-slice unit of t is one second. The prior probabilities of the *Action* node at t=0 and t=1 can be seen in Table 6 and Table 7, respectively.

Table 6. Prior probabilities of *Action* node at *t*=0.

Node states	Probabilities
KG	0.167
C1	0.167
C2	0.167
KP	0.167
LC	0.166
GT	0.166

Table 7. Prior probabilities of *Action* node at *t* >=1.

Node states	Prior state (*t*=0)	Probabilities (*t* >=1)
KG	KG	0.98
C1	C1	0.98
C2	C2	0.98
KP	KP	0.98
LC	LC	0.98
GT	GT	0.98

The probability of other node state combinations = 0.004

Furthermore, the *Predicted Action* node has the same states as the *Action* node. Besides getting input from the *Action* node, the *Predicted Action* node reads the state of the *Risk* node. The *Risk* node has four states, including *No Risk*, *Overtaking Speed Risks*, *Collision Risk*, and *Minimum Space Risk*. Their prior probabilities are set to 0.25. The main function of the *Risk* node is to override the prediction state given the condition of the *Action* node and give it a new *t*=0 value. Fig. 4 visualizes the temporal probability distribution of the *Prediction Action* node, given the state of the *Action* and *Risk* nodes at *t*=0 are KG and *No Risk*, respectively. The temporal probability distribution of the *Predicted Action* node is calculated based on a conditional probability table set for this node. The result indicates that *P(Predicted Action=KG) = 0.96* during *t*=0-3, which means that in the next three seconds, the predicted action of AA is 96% *keep going (KG)*. Similarly, Fig. 5 is the visualization of the temporal probability distribution of the *Predicted Action* node during *t*=0-3 given *P(Action=KG) = 1* and *P(Risk=OvertakingSpeedRisk) = 1 at t=0, respectively*. The result indicates that *P(Predicted Action=C1) = 0.96* during *t*=0-3.

The Carla instances are then replayed to see the performance of the DBN-based action predictor. Table 8 presents the performance comparison between the proposed method and the baseline method from [18]. The baseline method is selected because it also

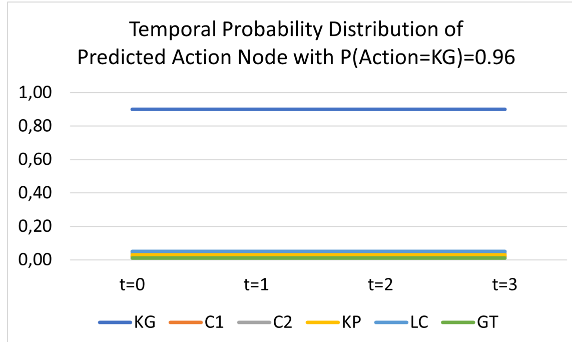

Figure 4. The visualization of the temporal probability distribution of the predicted action node during *t*=0-3 given *P(Action=KG) = 1* and *P(Risk=NoRisk) = 1 at t=0, respectively*.

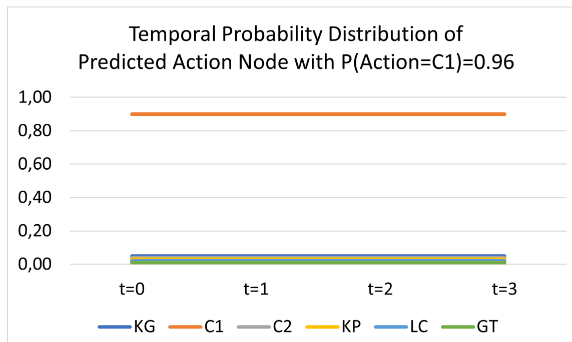

Figure 5. The visualization of the temporal probability distribution of the predicted action node during *t*=0-3 given *P(Action=KG) = 1* and *P(Risk=OvertakingSpeedRisk) = 1 at t=0, respectively.*

presents action classification in the driving context, even though it is only developed for lane change detection (similar to LC in this study). The other actions in this study (KG, $C1$, $C2$, KP, and GT) are not covered in the baseline method. Moreover, similar to the proposed method, the baseline method also presents non-contemporaneous dependency distinguishing variables within different periods.

The performance comparison is divided into two groups. In the first group, the comparison of action classifier at *t*=0 representing current action is presented. The second group presents the action classifier at $t+3$ representing AA's action in the next three seconds. The performance of the action (or maneuver) classifier at both *t*=0 and $t+3$ demonstrated by the proposed method outperformed the baseline method, reaching 98% and 67%, respectively. According to [18], the 67% of DBN accuracy is considered average and acceptable, particularly when the predictor is used in a dynamic and unpredictable

Table 8. Comparison to the baseline method from [18].

Maneuver	Current (t=0)		t+3	
Classification	A	B	A	B
KG	1.0	N/A	0.96	N/A
C1	1.0	N/A	0.96	N/A
C2	0.9	N/A	0.6	N/A
KP	0.76	N/A	0.5	N/A
LC	1.0	0.72	0.65	0.63
GT	0.69	N/A	0.4	N/A
Average	**0.98**	0.72	**0.67**	0.63

Notes:
A = Our proposed approach
B = Baseline method

environment. Even though this study only reached the average performance of accuracy in predicting immediate future actions, some benefits can still be obtained. For example, the human driver can use the results to compare their situational awareness with those from AA so a proper reaction can be made to anticipate improper AA's actions in the immediate future given various driving situations.

Two main factors degrade the level of accuracy. Firstly, before t=3, the state of the *Risk* node is changed, so that the predicted action is different from the state of the *Action* node at t=0. Secondly, the performance of the action classifier.

5. Conclusions

This paper aims to develop an approach to predict IA's immediate future actions. The proposed approach is based on Naive Bayes to develop IA's action classifier and DBN to predict IA's immediate future action. This research implements the proposed approach in a collaborative driving context with the overtaking scenario for the experimental case. The performance of DBN is highly tight to the accuracy of the classifier generated by the Naive Bayes. Lower accuracy of the Naive Bayes will supply wrong inputs for DBN to predict the AA's immediate future actions. Furthermore, future research can be directed to other scenarios from other contexts.

References

[1] N. Stanton, P. Salmon, G. Walker, E. Salas, and P. Hancock, "State-of-science: Situation awareness in individuals, teams and systems," *Ergonomics*, vol. 60, no. 4, pp. 449–466, 2017.

[2] M. Endsley, "From here to autonomy: Lessons learned from human–automation research," *Human Factors*, vol. 59, no. 1, pp. 5–27, 2017.

[3] W. Xu, "Toward human-centered ai: a perspective from human-computer interaction," *interactions*, vol. 26, no. 4, pp. 42–46, 2019.

[4] M. McNeese, M. Demir, N. Cooke, and C. Myers, "Teaming with a synthetic teammate: insights into human-autonomy teaming," *Human Factors*, vol. 60, no. 2, pp. 262–273, 2018.

[5] O. McAree, J. Aitken, and S. Veres, "Towards artificial situation awareness by autonomous vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7038–7043, 2017.

[6] O. Reynolds, "Towards model-driven self-explanation for autonomous decision-making systems," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. ACM, 2019, pp. 624–628.

[7] F. Ekman, M. Johansson, and S. J., "Creating appropriate trust in automated vehicle systems: A framework for hmi design," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 1, pp. 95–101, 2018.

[8] J. Fahndrich, S. Ahrndt, and S. Albayrak, "Towards self-explaining agents," in *Trends in Practical Applications of Agents and Multiagent Systems*, vol. 48. Springer, 2013, pp. 147–154.

[9] Y. Xing, C. Huang, and C. Lv, "Driver-automation collaboration for automated vehicles: a review of human-centered shared control," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1964–1971.

[10] L. Sanneman and J. Shah, "A situation awareness-based framework for design and evaluation of explainable ai," in *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*. Springer, 2020, pp. 94–110.

[11] C. Jansen, L. Boyle, L. Kun, W. Ju, and L. Chuang, "A hidden markov framework to capture human–machine interaction in automated vehicles," *International Journal of Human-Computer Interaction*, vol. 35, no. 11, pp. 947–955, 2019.

[12] J. Zhang, Y. Liu, P. Hansen, J. Wang, and F. You, "Interactive framework of cooperative interface for collaborative driving," in *HCI in Mobility, Transport, and Automotive Systems: Third International Conference, MobiTAS 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings*. Springer, 2021, pp. 351–364.

[13] K. Parsa, M. Hassall, and M. Naderpour, "Enhancing alarm prioritization in the alarm management lifecycle," *IEEE Access*, vol. 10, pp. 99–111, 2021.

[14] ——, "Process alarm modeling using graph theory: Alarm design review and rationalization," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2257–2268, 2020.

[15] M. Naderpour, J. Lu, and G. Zhang, "An abnormal situation modeling method to assist operators in safety-critical systems," *Reliability Engineering & System Safety*, vol. 133, pp. 33–47, 2015.

[16] ——, "An intelligent situation awareness support system for safety-critical environments," *Decision support systems*, vol. 59, pp. 325–340, 2014.

[17] A. Benterki, M. Boukhnifer, V. Judalet, and C. Maaoui, "Artificial intelligence for vehicle behavior anticipation: Hybrid approach based on maneuver classification and trajectory prediction," *IEEE Access*, vol. 8, pp. 56 992–57 002, 2020.

[18] J. Li, B. Dai, X. Li, X. Xu, and D. Liu, "A dynamic bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification," *Electronics*, vol. 8, no. 1, p. 40, 2019.

[19] H. Zhang, R. Fu, C. Wang, Y. Guo, and W. Yuan, "Turning maneuver prediction of connected vehicles at signalized intersections: A dictionary learning-based approach," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 23 142–23 159, 2022.

[20] R. Kridalukmana, H. Lu, and M. Naderpour, "Self-explaining abilities of an intelligent agent for transparency in a collaborative driving context," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 6, pp. 1155–1165, 2022.

[21] Y. Xing, C. Lv, D. Cao, and P. Hang, "Toward human-vehicle collaboration: Review and perspectives on human-centered collaborative automated driving," *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103199, 2021.

[22] D. Tran, J. Du, W. Sheng, D. Osipychev, Y. Sun, and H. Bai, "A human-vehicle collaborative driving framework for driver assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3470–3485, 2018.