automation

MDPI

Article

Multi-Level Feature Extraction and Classification for Lane Changing Behavior Prediction and POD-Based Evaluation

Zahra Rastin * and Dirk Söffker

Chair of Dynamics and Control, University of Duisburg-Essen, 47057 Duisburg, Germany; soeffker@uni-due.de
* Correspondence: zahra.rastin@uni-due.de

Abstract: Lane changing behavior (LCB) prediction is a crucial functionality of advanced driver-assistance systems and autonomous vehicles. Predicting whether or not the driver of a considered ego vehicle is likely to change lanes in the near future plays an important role in improving road safety and traffic efficiency. Understanding the underlying intentions behind the driver's behavior is an important factor for the effectiveness of assistance and monitoring systems. Machine learning (ML) algorithms have been broadly used to predict this behavior by analyzing datasets of traffic and driving data related to the considered ego vehicle. However, this technology has not yet been widely adopted in commercial products. Further improvements in these algorithms are necessary to enhance their robustness and reliability. In some domains, receiver operating characteristic and precision-recall curves are commonly used to evaluate ML algorithms, not considering the effects of process parameters in the evaluation, while it might be necessary to access the performance of these algorithms with respect to such parameters. This paper proposes the use of deep autoencoders to extract multi-level features from datasets, which can then be used to train an ensemble of classifiers. This allows for taking advantage of high feature-extraction capabilities of deep learning models and improving the final result using ensemble learning techniques. The concept of probability of detection is used in combination with the networks employed here to evaluate which classifiers can detect the correct LCB better in a statistical sense. Applications on data acquired from a driving simulator show that the proposed method can be adopted to improve the reliability of the classifiers, and ensemble ANNs perform best in predicting the upcoming human behavior in this dynamical context earlier than 3 s before the event itself.

Keywords: lane changing behavior prediction; machine learning; performance evaluation; probability of detection

Citation: Rastin, Z.; Söffker, D. Multi-Level Feature Extraction and Classification for Lane Changing Behavior Prediction and POD-Based Evaluation. *Automation* **2024**, *5*, 310–323. https://doi.org/10.3390/automation5030019

Academic Editor: Jahangir Hossain

Received: 22 May 2024
Revised: 8 July 2024
Accepted: 17 July 2024
Published: 22 July 2024

1. Introduction and Motivation

Every year, millions of people are killed or injured worldwide in traffic accidents that are mostly caused by human errors [1,2]. Advancements in vehicle, computer, and sensing technologies during the last decades have set the scene for emerging research on advanced driver-assistance systems (ADASs) and autonomous vehicles (AVs), aiming to provide safety and efficiency in traffic [3]. The development of AVs is still in the early stages, but it has the potential to significantly increase road safety by supporting the role of human drivers or assisting them to make better decisions [4,5]. Lane changing is a common driving behavior whose prediction is crucial for ADASs and AVs. Lane changing is deemed to be a complicated task that requires speed adjustment, finding an appropriate gap in the target lane, informing other drivers of lane changing intention, and controlling the dynamics of the vehicle at the same time. Drivers' lane changing behavior is closely linked to their driving style [6]. This behavior often negatively impacts traffic flow, leading to congestion, stop-and-go waves, and bottlenecks [7,8]. It is also a significant safety concern, with unsafe lane changes responsible for nearly 5% of all accidents worldwide [9]. Understanding and predicting lane changing behaviors (LCBs) is crucial for improving driver assistance and

warning strategies, thereby implicitly enhancing traffic safety. If the upcoming LCB can be predicted by ADASs, drivers can be guided accordingly to evaluate dangerous decisions and avoid unsafe behaviors [10]. Additionally, LCB prediction and modeling are essential in microsimulation to forecast how AVs impact urban network efficiency and road safety, emphasizing their important role in guiding AV deployment strategies [11,12].

Various mathematical approaches to LCB prediction have been proposed in the literature [13–17]. Recently, LCB prediction is usually performed using machine learning (ML) algorithms, which leads to developing more intelligent prediction models [18,19]. Among these algorithms, artificial neural networks (ANNs) support vector machines (SVMs), and hidden Markov models (HMMs) are commonly used to predict LCB [18]. Sharma et al. [20] combined an SVM and an HMM in a two-layer hierarchical model to predict LCB on highways. An SVM was used in the first layer to distinguish lane changing and lane keeping behavior; then, in the second layer, a continuous HMM incorporated with a Gaussian matrix model was employed to classify the output of the SVM to lane changing to left (LCL) or lane changing to right (LCR). Li et al. [21] introduced a combined neural network model for LCB prediction, employing gradient boosting decision trees to mine driving rules, convolutional neural networks to find spatial features, and long short-term memory networks to capture temporal features. Simulation results demonstrated enhanced prediction accuracy. Deng and Söffker [22] presented a strategy for improving the reliability of ML algorithms utilized in LCB prediction. Input features were selected using a prefilter. The parameters that are conventionally set manually prior to training ML models, as well as the prefilter thresholds, could be determined using nondominated sorting genetic algorithm II to enhance the performance of ML algorithms. Dou et al. [23] developed a model based on a combination of SVMs and ANNs to predict whether it is feasible and suitable to change lanes considering certain environmental conditions. It was shown that the developed model is more accurate as compared with Bayes classifiers and decision trees. In a study conducted by Zheng et al. [24], an ANN was developed to predict lane changing decisions utilizing trajectory data collected on a highway using video cameras. A sensitivity analysis was performed to assess the effect of highway vehicles on LCB. The findings indicated that ANNs demonstrated satisfactory performance in terms of accuracy, while also providing a quantitative assessment of the influence of heavy vehicles on lane changing decisions made by car drivers. More details on ML-based models concerning the prediction and recognition of lane changing behavior are given in ref. [18].

While the potential of driver intention recognition technology is acknowledged, not many commercial products have been developed using this technology in practice. The performance of ML algorithms needs to be further improved to increase their robustness and reliability [25]. Additionally, there is a requirement for appropriate approaches to assess ML algorithms' performance effectively, allowing ongoing improvement. Ensemble learning is one of the most successful techniques for model improvement that can enhance the overall robustness by combining multiple light models [25]. For example, Zhang and Fu [26] proposed an ensemble K-nearest neighbor algorithm for recognizing vehicle behavior that was about 8% more accurate than the base classifier in similar conditions. Additionally, using suitable features as inputs to ML models is essential to guarantee successful network operation. Furthermore, it is crucial to select an optimal set of hyperparameters for ML networks, as they directly impact the networks' performance [27].

To ensure the reliability of ML-based approaches, different metrics have been used for model validation. For instance, Sharma et al. [20] and Wang et al. [28] validated their proposed models according to their accuracy. Dang et al. [29] used a modified F-score as the evaluation metric. Receiver operating characteristic (ROC) and precision-recall (PR) curves are commonly used for evaluating ML algorithms [30]. For example, in a study conducted by Mozaffari et al. [31], ROC curves were employed to validate the proposed ML-based LCB prediction algorithm. These curves might lead to a wrong performance evaluation when used with unbalanced datasets; PR curves are alternatively used to tackle this problem [32,33].

None of the above-mentioned techniques consider the effects of process parameters in evaluation directly. Process parameters (different from a network's hyperparameters) are the parameters of the task that influence the recognizability and the final result, such as the size of the damage in a damage detection task. Despite their importance, such effects have not been given enough attention in assessing ML-based approaches [34]. Ameyaw et al. [35] used the probability of detection (POD) approach to evaluate and compare the reliabilities of ML models utilized for LCB prediction, considering the time remaining until the lane changing event as the process parameter. The considered classifiers are ANNs, SVMs, HMMs, random forests (RFs), and improved versions of these algorithms (developed in ref. [22]). A noise analysis was adopted to include false-positive values in the comparison of classifiers.

This paper uses deep neural networks to extract features required for training ML algorithms popularly used for LCB prediction. A deep autoencoder (DAE) is used for feature extraction from a dataset consisting of traffic and driving data related to an ego vehicle. Multi-level features extracted from different layers of the DAE are used to train an ensemble of classifiers of one kind. Features from each layer might contain useful information about the data and their hidden patterns and the likelihood of lane changing; therefore, the final result is obtained considering both high-level and low-level features. Genetic algorithms (GAs) are employed to tune the hyperparameters of the models. The idea proposed here is to use the previously developed POD-based approach to assess the performance of the classifiers, taking the effects of process parameters into account for evaluation. To compare the results with those obtained from a study conducted by Ameyaw et al. [35], the same process parameter and classifiers (ANNs, SVMs, HMMs, and RFs) are considered here. Applications on data from a driving simulator, including three classes of LCB (LCL, LCR, and lane keeping), show that the combination of multi-level features, ensemble learning, and hyperparameter optimization enhances the ability of the classifiers to predict the true LCB, and ensemble ANNs perform best in predicting the upcoming human behavior in this dynamical context earlier than 3 s before the event itself.

This paper is organized as follows: After the brief introduction, a concise overview of the utilized algorithms and the POD approach is presented in Section 2. In Section 3, the proposed methodology for LCB prediction and model evaluation is explained. The results of applying this methodology are presented and discussed in Section 4. Finally, in Section 5, the proposed approach and the main findings are summarized, and conclusions are drawn.

2. Theoretical Background

As mentioned previously, among various ML algorithms, ANN-, SVM-, and HMM-based approaches are frequently employed for LCB prediction. Unlike well-known ANN and SVM approaches, HMMs capture dependencies between data points over time, and are specialized to deal with sequential data [36]. Hidden Markov models are statistical models that analyze sequences of observable data (here, driving data) that are associated with underlying hidden states. The hidden states are not directly observable; for example, in the present task, hidden states are the correct LCBs. The goal of an HMM is to predict the most probable sequence of hidden states that lead to the observed data [37].

Ensemble learning is a technique in ML that tries to achieve a more accurate and robust result by combining the results obtained from multiple base models to enhance the overall performance. Various methods can be taken to merge individual model outputs. Voting, averaging, stacking, and winner-take-all are examples of commonly used strategies [38]. Random forest is a popular ensemble classifier that fuses the outputs of several decision trees through a voting process for classification tasks. A decision tree is a model consisting of a root node that initiates the tree, internal nodes that are decision points leading to splits, and leaf nodes distinguishing different classes in classification problems [39].

The performance of all of the applied approaches (ANNs, SVMs, HMMs, and RFs) highly depends on the set of features used for training. The quality and relevance of these features significantly influence the efficacy of the models. Autoencoders can be used

effectively to extract features needed for training traditional ML algorithms. These networks are a type of unsupervised ANN, mainly used for data compression and consisting of an encoder part that reduces the size of the input data and a decoder part that rebuilds the input data using the compressed output of the encoder. The network learns to keep only the main features of the input data in the output layer of the encoder trying to rebuild the data in the decoder part.

Genetic algorithms can be employed to select the most suitable combination of hyperparameters for ML networks. It is a curtail task as these parameters have a direct influence on the performance of the networks. Considering this, GAs simulate the process of natural selection by evolving a population of potential solutions (here, hyperparameter sets) across generations. During this process, the algorithm explores a large search space and finds the near-optimal solution for the considered problem.

The POD approach was initially developed to validate nondestructive evaluation (NDE) methods. It can be utilized to evaluate the effectiveness of detection systems considering the inherent variability and uncertainty associated with the detection process [40]. This approach is gaining interest in other domains where it was previously less commonly applied, such as the structural health monitoring field and the nuclear industry [41]. The POD evaluation results in a so-called POD curve. In the field of NDE, such a curve depicts the likelihood of detecting a defect as a function of defect size [42,43]. Here, the outputs of the ensemble classifiers are used to obtain POD curves, showing the probability of detecting the correct LCB as a function of the time remaining until the event (as process parameter) [35]. The data utilized to generate POD curves include either binary statements indicating the presence or absence of a target (hit/miss) or signal response data, providing a continuous quantitative assessment of a target (â versus a) [43,44]. Continuous data will be used for producing POD curves in this paper.

3. Methodology

The methodology suggested for predicting LCB and evaluating models is outlined in this section. The proposed approach when employing binary ANNs as classifiers to predict lane changing to right or left is depicted in Figure 1. The first step in the proposed method is to extract the features needed as inputs to ML algorithms using an autoencoder. Data including information regarding nearby vehicles and the condition of the considered ego vehicle, recorded every 0.05 s, are normalized between 0 and 1 and used to train the autoencoder. The utilized dataset is described further in Section 4. The encoder part of the autoencoder comprises an input layer with 34 neurons and 4 hidden layers with 24, 16, 8, and 4 neurons, respectively. The decoder layers are the inverse of the encoder layers. All layers use a Rectified Linear Unit (ReLU) activation function, with the exception of the final one, which employs the Sigmoid function. Mean Square Error (MSE) is selected as the loss function and minimized via an Adam optimizer. Genetic algorithm is used to obtain optimal values of batch size, epoch, and learning rate that minimizes the reconstruction error.

Multi-level features extracted from the four dense layers of the encoder are input to the ML algorithms. Three classes of LCB are regarded: LCL, LCR, and lane keeping. Considering ANN as an example, features from each layer are utilized to train two ANNs, each one considering LCL or LCR as the positive class and the two other classes as the negative ones. The same is done for the three other algorithms, resulting in a total of 32 trained models. Hyperparameters of the models are selected using GA to minimize the objective function

$$f = (1 - ACC) + (1 - DR) + FAR, \quad (1)$$

where ACC, DR, and FAR denote the accuracy, detection rate, and false alarm rate, respectively. By combining these three conflicting aspects, the GA is guided to define an optimal solution between high ACC/DR and false alarms, which is crucial for an effective model performance.

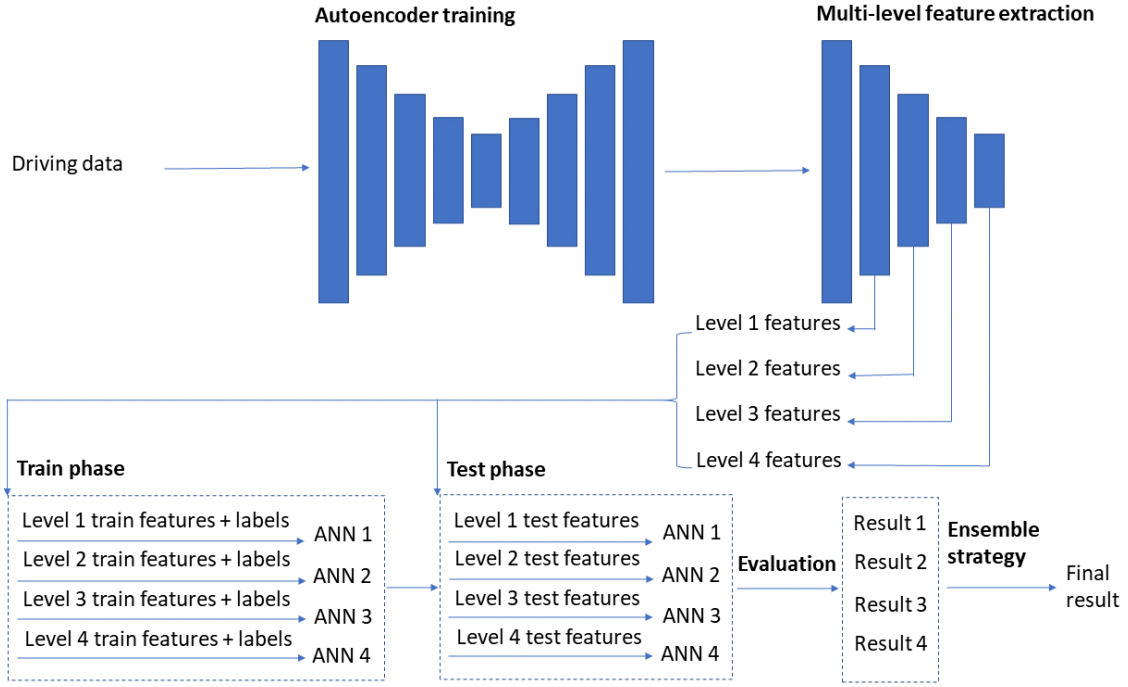

Figure 1. Lane changing behavior prediction using binary ANNs as classifiers.

The performance of the trained models in LCB prediction given test data is evaluated using the POD approach. The mh1823 POD software (version 7.3.4) [45] is used for this purpose. Features extracted from test data are fed to the trained models, and the probability that each sample of the data belongs to the positive class is extracted from the models. Time periods from 7 s prior to the lane change until the moment the event occurs are taken into account for evaluation, and the average of the obtained probabilities at each time point during this 7 s period is used as the response value for the POD-based performance assessment. The process for obtaining a POD is comprehensively explained in Military Handbook 1823A (MIL-HKBK-1823A) [43]. The main stages of this process for the purpose outlined in this paper are shown in Figure 2. As the first step in the POD-based evaluation of each classifier, the predicted probabilities ($\hat{a}$) are plotted as a function of time (a). Four possible combinations of logarithmic and Cartesian $\hat{a}$ and a axes are considered. The graph best satisfying the following criteria is chosen:

1. The data should seem to be well described by a straight line.
2. The variance must be uniform about the regression line.
3. The observations must the uncorrelated.

Regression analysis is conducted on data from the best graph using the maximum likelihood method (Figure 2a). Considering the $\hat{a}$ vs. a case, a line describing the data can be obtained as

$$\hat{a} = b + ma + \epsilon, \tag{2}$$

where b and m are the regression coefficients of the model and $\epsilon \sim N(0, \tau)$ is the corresponding error term following a normal distribution with a zero mean and a standard deviation equal to τ. To obtain the likelihood of reliable LCB prediction at each time point, the probability that the classifier will produce a response value above an arbitrary threshold, known as decision threshold ($\hat{a}_{th}$), must be determined. This probability with 50% confidence is equal to the area under the probability density function (PDF) obtained from the distribution of the error term in equation 2 and centered around the regression line, above $\hat{a}_{th}$ (the shaded area in Figure 2b). This probability at each time point is obtained as [43,46]

$$POD(a) = P(\hat{a} > \hat{a}_{th}) = 1 - P(\hat{a} \leq \hat{a}_{th}) = 1 - \phi\left(\frac{\hat{a}_{th} - (b + ma)}{\tau}\right), \tag{3}$$

where ϕ is the cumulative standard normal distribution function. To account for the uncertainty associated with estimating the parameters of the regression line, the Wald method is used to construct the 95 percentile POD curve. This curve represents the boundary beneath which 95% of the average POD curves would lie, if the study was repeated numerous times. A POD curve and its 95% lower confidence bound are shown in Figure 2c. In this figure, a_{90} and $a_{90/95}$ show time points when the POD and the associated 95% lower bound reach 90%.

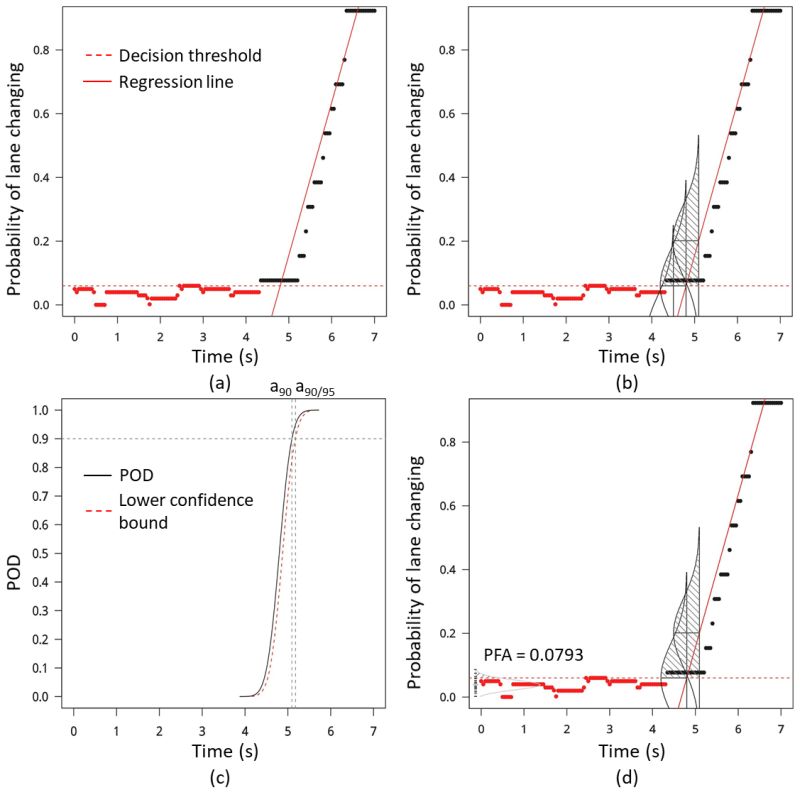

Figure 2. The main stages in the POD-based performance evaluation of the classifiers: (**a**) fitting a line to the data through regression analysis, (**b**) obtaining the POD at each time point with 50% confidence, (**c**) generating the POD curve and its 95% lower confidence bound, and (**d**) analyzing noise and computing false alarm probability.

As part of the POD-based evaluation, noise analysis is conducted to account for the probability of false alarm (PFA). Noise in this context is defined as response values that are random with respect to the process parameter and contain no useful target characterization information. In this study, the noise can be effectively represented by a Gaussian distribution. The PFA for each classifier is equal to the area under the noise PDF above $\hat{a}_{th}$ as

$$PFA = P(\hat{a}_{noise} > \hat{a}_{th}). \quad (4)$$

A target-response model with PDFs used for obtaining the POD curve and the PFA are shown in Figure 2d. Reducing $\hat{a}_{th}$ enhances the POD curve, but inevitably results in an increase in the PFA. The $\hat{a}_{th}$ for each classifier is chosen considering the trade-off between the POD and the PFA.

A winner-take-all ensemble strategy is applied to performance evaluation results from classifiers of one type having the same task. According to this strategy, multiple classifiers compete against each other; the ensemble output is finally taken from the classifier that achieves the best classification performance. For example, the final $a_{90/95}$ result of using ANN to predict LCR will be equal to the result from the fastest network among the four ANNs trained on features from different autoencoder layers (the lowest $a_{90/95}$), when

associated false alarm probabilities are equal. According to evaluation results, the most reliable model for an LCB prediction task can be specified.

4. Application and Results

The methodology is applied for LCB prediction using data acquired from a SCANeR™ studio driving simulator (Figure 3), also described in previous publications [30,35]. The simulator employs virtual sensors like cameras, radar, and lasers to gather data, enabling a thorough comprehension of the vehicle's surrounding. The driving environment is a highway with two lanes in each direction and simulated traffic conditions. Three drivers aged between 25 and 38, all possessing valid drivers' licenses, were recruited to gather the required data. Each participant drove for about 40 min to obtain the training dataset. Another 10 min of driving data were also recorded for the test phase. During driving, participants were allowed to overtake the preceding vehicle when it drove slowly, with the option to return to the initial lane afterward. The driving environment and lane changing/keeping behaviors are illustrated in Figure 4. In the driving simulation, the vehicle's lane is identified by the position of its center point. Lane changing is detected when this lane value shifts. The onset of a lane changing event is marked by the last notable change in the steering wheel angle. The interval between this steering adjustment and the moment when the lane value shifts is defined as the lane changing period.

Figure 3. SCANeR™ studio, Chair of Dynamics and Control, University of Duisburg–Essen, Germany.

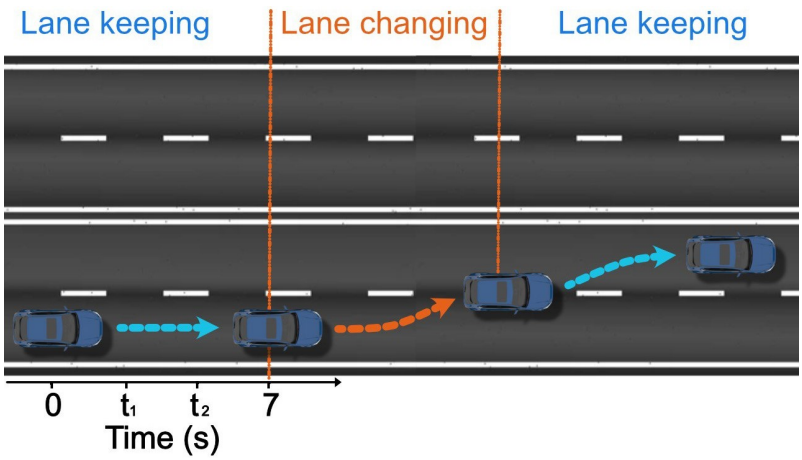

Figure 4. The driving environment and lane changing/keeping behaviors. In the considered 7 s period before a lane changing event, the earlier an algorithm predicts the LCB (at time t_1 as compared with t_2), the better the performance.

Samples of training and test datasets were recorded every 0.05 s and consisted of 26 observation variables, including the velocity of the considered ego vehicle and the surrounding vehicles, the distance between the ego vehicle and the surrounding vehicles, the time to collision to surrounding vehicles, the lane number, the turn signal indicator's

state, the engaged gear, the steering wheel angle, the heading angle, the accelerator pedal position, and the brake pedal pressure related to the ego vehicle. Categorical variables without intrinsic order were one-hot-encoded to prevent potential misinterpretation of ordinal relationships by algorithms. No data preprocessing was undertaken prior to feeding samples to the autoencoder.

The acquired data are used to train the autoencoder described in Section 3, and the multi-level features extracted from this network are input to ANN, SVM, HMM, and RF models. For each task (LCL or LCR prediction), four models of each type are trained, each one on features from a different autoencoder layer. Performance evaluation is conducted using test data, and the best model among the four is selected for the specified task (winner-take-all ensemble strategy). POD-based performance evaluation results in $a_{90/95}$ and PFA values. To compare the algorithms and specify the most suitable one for all of the considered models, the decision threshold ($\hat{a}_{th}$) is chosen so that the resulting PFA is less than 1%. The less the $a_{90/95}$ value, the earlier the algorithm predicts the LCB, and the better the performance. For example, in Figure 4, in the considered 7 s period before a lane changing event, the ML algorithm capable of predicting the LCB at time t_1 outperforms an approach predicting the LCB at time t_2. As shown later in this section, in the best case, the correct LCB is predicted about 7 s before the event ($a_{90/95} = 0.542$ s, so the algorithm predicts the LCB $7 - 0.542 = 6.458$ s before the event). Therefore, 7 s periods were chosen to show the results of the performance evaluation.

Considering the HMM trained on features from the second autoencoder layer for LCL prediction when fed with features from the first driver's test data as an example, the graphically illustrated results of the four combinations of logarithmic and Cartesian $\hat{a}$ (predicted probabilities belonging to LCL class for test data samples) and a (time points in the considered 7 s period) axes are depicted in Figure 5. In this example,the graph shown in Figure 5b meets the criteria mentioned in the previous section the best and, therefore, is chosen for conducting regression analysis.

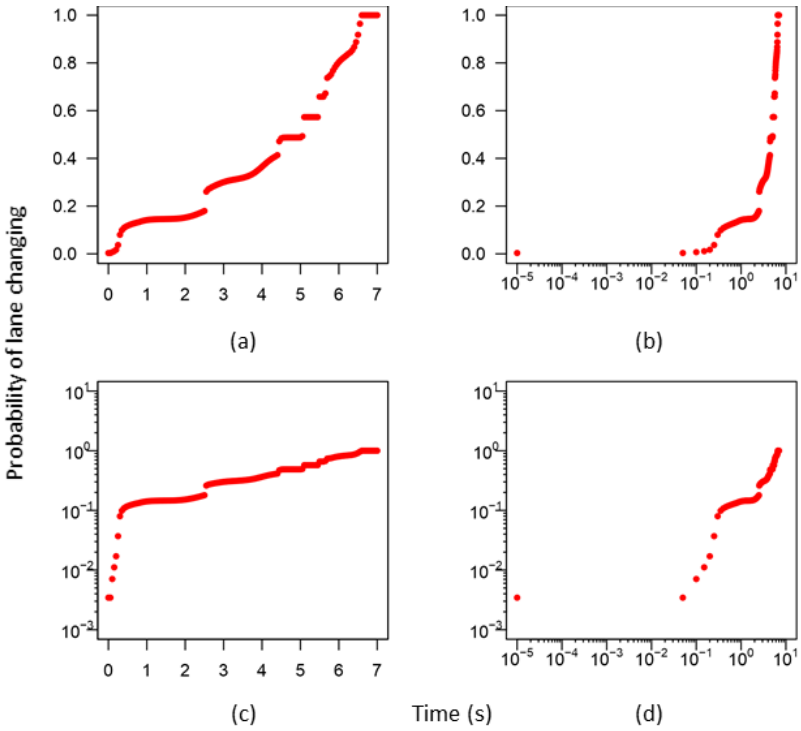

Figure 5. (**a**) $\hat{a}$ vs. a, (**b**) $\hat{a}$ vs. $\log(a)$, (**c**) $\log(\hat{a})$ vs. a, and (**d**) $\log(\hat{a})$ vs. $\log(a)$ models for LCL prediction using the HMM trained on features from the second autoencoder layer and test data from the first driver.

The obtained regression line is shown in Figure 6a. The decision threshold is set to 0.02, which results in a PFA equal to 0 (Figure 6b). The final POD curve and its 95% lower confidence bound are depicted in Figure 6c. According to this figure, $a_{90/95}$ for the considered HMM model is equal to 1.752 s.

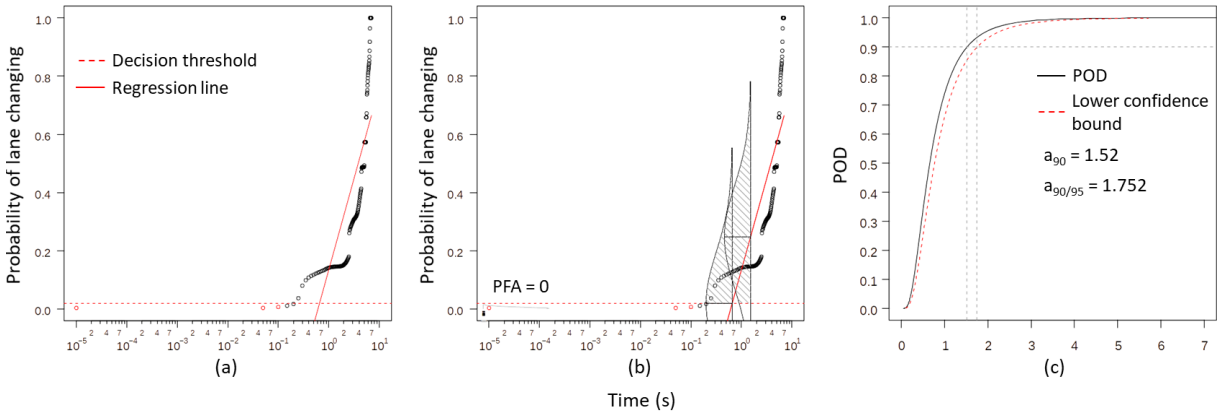

Figure 6. Performance evaluation for the HMM trained on features from the second autoencoder layer to predict LCL when fed with test data from the first driver: (**a**) the regression line, (**b**) the noise analysis result, and (**c**) the POD curve and its lower confidence bound.

Hidden Markov model performance evaluation results when trained on features from different layers and tested on the first driver's 10 min test data (different from the data used in the training phase) are given in Table 1. Using a winner-take-all ensemble strategy, the final $a_{90/95}$ result is 1.752 s, which is obtained from the HMM trained on the second layer features. This example shows that the best outcomes are not necessarily achieved from high-level features; therefore, employing multi-level features to obtain the most favorable result can be beneficial.

Table 1. Hidden Markov model performance evaluation results using the first driver's test data.

Related Encoder Layer No.	$a_{90/95}$ [S]
1	>7
2	1.752
3	1.87
4	1.786

The $a_{90/95}$ values from the considered models when trained to predict LCL/LCR and tested on data from the three drivers separately are presented in Tables 2 and 3. The related autoencoder layers are also included in the tables. Additionally, the overall FARs (independent from POD analysis) are given in these tables, showing that the algorithms perform well outside the considered 7 s periods and do not misclassify many samples of lane keeping data as LCL/LCR. The lowest $a_{90/95}$ values for LCL prediction are 1.22 s, 1.968 s, and 2.905 s for drivers 1, 2, and 3, respectively. These values are obtained from SVM for driver 1 and ANN for drivers 2 and 3. Additionally, the lowest $a_{90/95}$ values for LCR prediction are obtained from ANN for drivers 1 and 2 and SVM for driver 3 and are equal to 2.823 s, 3.211 s, and 4.355 s, respectively. It can be concluded that, among the considered models, ANNs are the most reliable algorithms for LCB prediction, followed by SVMs. Furthermore, in almost all cases, the best ANN results are achieved using features from the third autoencoder layer.

Table 2. Performance evaluation for LCL prediction.

Algorithm	Driver No.	$a_{90/95}$ (s)	Related Layer	Overall FAR
ANN	1	1.558	3	0.03
	2	1.968 *	3	
	3	2.905 *	4	
SVM	1	1.22 *	4	0.025
	2	2.069	3	
	3	2.971	3	
HMM	1	1.752	2	0.052
	2	2.701	2	
	3	3.533	2	
RF	1	5.702	2	0
	2	3.471	1	
	3	2.975	3	

* The best $a_{90/95}$ value for each driver.

Table 3. Performance evaluation for LCR prediction.

Algorithm	Driver No.	$a_{90/95}$ (s)	Related Layer	Overall FAR
ANN	1	2.823 *	3	0.065
	2	3.211 *	3	
	3	4.752	3	
SVM	1	4.889	1	0.021
	2	3.361	1	
	3	4.355 *	2	
HMM	1	4.166	3	0.037
	2	3.839	3	
	3	4.746	2	
RF	1	5.843	4	0.0035
	2	4.445	4	
	3	5.339	4	

* The best $a_{90/95}$ value for each driver.

To compare the reliability of the methodology presented here with that of the improved ANN, SVM, HMM, and RF developed in ref. [22], the $a_{90/95}$ values are calculated again, this time considering the DR at each time step as the response value in a POD-based evaluation. Just as before, decision thresholds are selected so that the noise analysis results in PFA values of less than 1%. For an appropriate comparison, the results presented in ref. [35] are also recalculated to set false alarm probabilities to less than 1%. The new results are presented in Tables 4 and 5 for LCL and LCR prediction. According to these tables, in almost all cases, the method proposed here improves the reliability of the classifiers. In only three cases, the new methodology yields $a_{90/95}$ values about 4 s greater than the values obtained from the previous models (shown in red in Table 4). Again, ANNs have the best performance among the considered algorithms, achieving the best results in four of six cases.

Comparing the results presented in Tables 2 and 3 with those presented in Tables 4 and 5, it can be seen that, in general, using DR as the response value results in higher $a_{90/95}$ amounts and can be considered as a more conservative way of POD-based evaluation. However, using the probabilities extracted from the algorithms directly provides more unaffected results. The POD curves obtained for ANNs trained on third autoencoder layer features, when using test data from the three drivers together, is shown in Figure 7. Accordingly, $a_{90/95}$ is less than 4 s for these networks, so ANNs can reliably predict LCBs earlier than $7 - 4 = 3$ s before the event.

Table 4. Comparison between the performance of the method proposed here and that of the method presented in previous studies [22] for LCL prediction.

Algorithm	Driver No.	Current $a_{90/95}$ (s)	Previous $a_{90/95}$ (s)
ANN	1	0.542 *	6.676
	2	1.749 *	6.62
	3	4.849 *	5.72
SVM	1	2.541	6.183
	2	2.36	5.389
	3	6.527	>7
HMM	1	1.655	6.231
	2	2.661	3.473
	3	5.725 **	1.248
RF	1	6.879 **	3.181
	2	4.712	6.005
	3	>7 **	5.631

* The best $a_{90/95}$ value for each driver, ** worse performance as compared with the previous models.

Table 5. Comparison between the performance of the method proposed here and that of the method presented in previous studies [22] for LCR prediction.

Algorithm	Driver No.	Current $a_{90/95}$ (s)	Previous $a_{90/95}$ (s)
ANN	1	3.162 *	6.347
	2	4.711	5.823
	3	4.777	6.039
SVM	1	5.081	6.24
	2	4.393	5.006
	3	4.369 *	6.404
HMM	1	4.219	5.644
	2	4.001 *	5.908
	3	4.764	6.006
RF	1	6.138	6.942
	2	4.786	5.07
	3	5.524	6.298

* The best $a_{90/95}$ value for each driver.

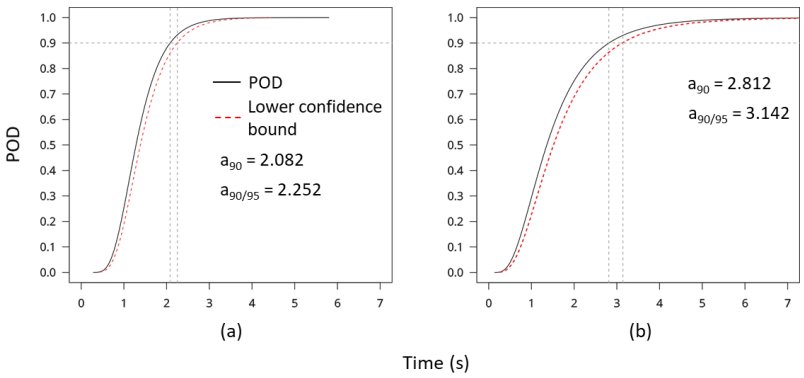

Figure 7. Probability of detection curves related to ANNs trained on third autoencoder layer features, when using test data from all drivers together, for (**a**) LCL and (**b**) LCR prediction and the obtained $a_{90/95}$ values.

5. Summary, Conclusions, and Outlook

This paper introduces a method aimed at improving the reliability and robustness of ML algorithms used to predict the LCB of the driver of a considered ego vehicle and the appropriate performance evaluation. The method utilizes multi-level features extracted

from various layers of a DAE to train an ensemble of classifiers of the same type. The resulting output for each type of classifier is determined by considering both high-level and low-level features using a winner-take-all ensemble strategy. Artificial neural networks, SVMs, HMMs, and RFs are employed as classifiers. The hyperparameters of all models are optimized using GAs.

The performance of the classifiers is evaluated using a POD-based approach. Unlike classical performance evaluation metrics (e.g., DR and ACC), as well as ROC and PR curves, which are commonly used for evaluating ML algorithms, the POD-based approach takes the effects of process parameters into account for the evaluation and provides new insights into the reliability of ML algorithms. The process parameter considered here is the remaining time until the lane changing event, and the classifiers' capability in predicting the correct intended driver behavior is evaluated relative to the remaining time before the event occurs.

The proposed methodology is validated using data from a driving simulator, including three classes of LCB: LCL, LCR, and lane keeping. A comparison with previous studies demonstrates that the combination of multi-level features, ensemble learning, and hyperparameter optimization improves the classifiers' performance and ensemble ANNs perform best, predicting the true LCB earlier than 3 s before the event occurs.

In future research, the hit/miss version of the POD approach could be applied and modified by taking into account the probability of hit and miss data acquired from ML algorithms, depending on the process parameter. Additionally, exploring the potential of optimizing the hyperparameters of ML algorithms to enhance their performance by minimizing the results of POD analysis could be investigated.

Author Contributions: The contributions of the authors to this work are as follows: Conceptualization, D.S. and Z.R.; methodology, D.S. and Z.R.; software, Z.R.; validation, D.S. and Z.R.; formal analysis, Z.R.; investigation, Z.R.; resources, D.S.; writing—original draft preparation, Z.R.; writing—review and editing, Z.R. and D.S.; visualization, D.S.; supervision, D.S.; project administration, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We gratefully acknowledge the support provided by the Open Access Publication Fund of the University of Duisburg–Essen.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. World Health Organization (WHO). Available online: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (accessed on 16 May 2024).
2. Bucsuházy, K.; Matuchová, E.; Zvala, R.; Moravcová, P.; Kostíková, M.; Mikulec, R. Human factors contributing to the road traffic accident occurrence. *Transp. Res. Procedia* **2020**, *45*, 555–561. [CrossRef]
3. Mahajan, V.; Katrakazas, C.; Antoniou, C. Prediction of lane-changing maneuvers with automatic labeling and deep learning. *Transp. Res. Rec.* **2020**, *2674*, 336–347. [CrossRef]
4. Stoma, M.; Dudziak, A.; Caban, J.; Droździel, P. The future of autonomous vehicles in the opinion of automotive market users. *Energies* **2021**, *14*, 4777. [CrossRef]
5. Wang, J.; Zhang, L.; Huang, Y.; Zhao, J.; Bella, F. Safety of autonomous vehicles. *J. Adv. Transp.* **2020**, *2020*, 1–13. [CrossRef]
6. Chu, H.; Zhuang, H.; Wang, W.; Na, X.; Guo, L.; Zhang, J.; Gao, B.; Chen, H. A Review of Driving Style Recognition Methods From Short-Term and Long-Term Perspectives. *IEEE Trans. Intell. Veh.* **2023**, *8*, 4599–4612. [CrossRef]
7. Ali, Y.; Hussain, F.; Bliemer, M.C.; Zheng, Z.; Haque, M.M. Predicting and explaining lane-changing behaviour using machine learning: A comparative study. *Transp. Res. Part C Emerg. Technol.* **2022**, *145*, 103931. [CrossRef]
8. Ali, Y. Investigation of Lane-Changing Behaviour in a Connected Environment. Ph.D. Thesis, University of Queensland, Queensland, Australia, 10 April 2020.
9. Hou, Y.; Edara, P.; Sun, C. Situation assessment and decision making for lane change assistance using ensemble learning methods. *Expert Syst. Appl.* **2015**, *42*, 3875–3882. [CrossRef]

10. Deng, Q.; Wang, J.; Hillebrand, K.; Benjamin, C.R.; Söffker, D. Prediction performance of lane changing behaviors: A study of combining environmental and eye-tracking data in a driving simulator. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3561–3570. [CrossRef]
11. Pelizza, A.; Orsini, F.; Yilmaz-Niewerth, S.; Rossi, R.; Friedrich, B. Exploring the impact of automated vehicles lane-changing behavior on urban network efficiency. In Proceedings of the 2023 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Nice, France, 14–16 June 2023; pp. 1–6.
12. Berrazouane, M.; Tong, K.; Solmaz, S.; Kiers, M.; Erhart, J. Analysis and Initial Observations on Varying Penetration Rates of Automated Vehicles in Mixed Traffic Flow utilizing SUMO. In Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), Graz, Austria, 4–8 November 2019; pp. 1–7.
13. Do, J.; Han, K.; Choi, S.B. Lane change–intention inference and trajectory prediction of surrounding vehicles on highways. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3813–3825. [CrossRef]
14. Dong, C.; Dolan, J.M. Continuous behavioral prediction in lane-change for autonomous driving cars in dynamic environments. In Proceedings of the 2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3706–3711.
15. Talebpour, A.; Mahmassani, H.S.; Hamdar, S.H. Modeling lane-changing behavior in a connected environment: A game theory approach. *Transp. Res. Procedia* **2015**, *7*, 420–440. [CrossRef]
16. Shi, W.; Zhang, Y.P. Decision analysis of lane change based on fuzzy logic. *Appl. Mech. Mater.* **2013**, *419*, 790–794. [CrossRef]
17. Baumann, M.; Krems, J.F. Situation awareness and driving: A cognitive model. In *Modelling Driver Behaviour in Automotive Environments: Critical Issues in Driver Interactions with Intelligent Transport Systems*; Cacciabue, P.C., Ed.; Springer: London, UK, 2007; pp. 253–265.
18. David, R.; Söffker, D. A review on machine learning-based models for lane-changing behavior prediction and recognition. *Front. Future Transp.* **2023**, *4*, 950429. [CrossRef]
19. Wei, C.; Hui, F.; Khattak, A.J. Driver lane-changing behavior prediction based on deep learning. *J. Adv. Transp.* **2021**, *2021*, 1–15. [CrossRef]
20. Sharma, O.; Sahoo, N.; Puhan, N. Highway lane-changing prediction using a hierarchical software architecture based on support vector machine and continuous hidden markov model. *Int. J. Intell. Transp. Syst. Res.* **2022**, *20*, 519–539. [CrossRef]
21. Li, R.; Shu, X.; Li, C. Driving Behavior Prediction Based on Combined Neural Network Model. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 4488–4496. [CrossRef]
22. Deng, Q.; Söffker, D. Classifying human behaviors: Improving training of conventional algorithms. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 1060–1065.
23. Dou, Y.; Yan, F.; Feng, D. Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. In Proceedings of the 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, Canada, 12–15 July 2016; pp. 901–906.
24. Zheng, J.; Suzuki, K.; Fujita, M. Predicting driver's lane-changing decisions using a neural network model. *Simul. Model. Pract. Theory* **2014**, *42*, 73–83. [CrossRef]
25. Xing, Y.; Lv, C.; Wang, H.; Cao, D.; Velenis, E. An ensemble deep learning approach for driver lane change intention inference. *Transp. Res. Part C Emerg. Technol.* **2020**, *115*, 102615. [CrossRef]
26. Zhang, H.; Fu, R. An ensemble learning—Online semi-supervised approach for vehicle behavior recognition. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 10610–10626. [CrossRef]
27. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [CrossRef]
28. Wang, W.; Qie, T.; Yang, C.; Liu, W.; Xiang, C.; Huang, K. An intelligent lane-changing behavior prediction and decision-making strategy for an autonomous vehicle. *IEEE Trans. Ind. Electron.* **2021**, *69*, 2927–2937. [CrossRef]
29. Dang, H.Q.; Fürnkranz, J.; Biedermann, A.; Hoepfl, M. Time-to-lane-change prediction with deep learning. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–7.
30. Ameyaw, D.A.; Deng, Q.; Söffker, D. How to evaluate classifier performance in the presence of additional effects: A new POD-based approach allowing certification of machine learning approaches. *Mach. Learn. Appl.* **2022**, *7*, 100220. [CrossRef]
31. Mozaffari, S.; Arnold, E.; Dianati, M.; Fallah, S. Early lane change prediction for automated driving systems using multi-task attention-based convolutional neural networks. *IEEE Trans. Intell. Veh.* **2022**, *7*, 758–770. [CrossRef]
32. Giglioni, V.; García-Macías, E.; Venanzi, I.; Ierimonti, L.; Ubertini, F. The use of receiver operating characteristic curves and precision-versus-recall curves as performance metrics in unsupervised structural damage classification under changing environment. *Eng. Struct.* **2021**, *246*, 113029. [CrossRef]
33. Guo, H.; Keyvan-Ekbatani, M.; Xie, K. Lane change detection and prediction using real-world connected vehicle data. *Transp. Res. Part C Emerg. Technol.* **2022**, *142*, 103785. [CrossRef]
34. Ameyaw, D.A.; Deng, Q.; Söffker, D. Probability of detection (POD)-based metric for evaluation of classifiers used in driving behavior prediction. In Proceedings of the Annual Conference of the PHM Society, Scottsdale, AZ, USA, 22 September 2019; pp. 23–26.

35. Ameyaw, D.A.; Deng, Q.; Söffker, D. Evaluating machine learning-based classification approaches: A new method for comparing classifiers applied to human driver prediction intentions. *IEEE Access* **2022**, *10*, 62429–62439. [CrossRef]
36. Narimatsu, H.; Kasai, H. Duration and interval hidden Markov model for sequential data analysis. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–8.
37. Mor, B.; Garhwal, S.; Kumar, A. A systematic review of hidden Markov models and their applications. *Arch. Comput. Methods Eng.* **2021**, *28*, 1429–1448. [CrossRef]
38. Yang, Y.; Lv, H.; Chen, N. A survey on ensemble learning under the era of deep learning. *Artif. Intell. Rev.* **2023**, *56*, 5545–5589. [CrossRef]
39. Sharma, H.; Kumar, S. A survey on decision tree algorithms of classification in data mining. *Int. J. Sci. Res.* **2016**, *5*, 2094–2097.
40. Ameyaw, D.A.; Rothe, S.; Söffker, D. A novel feature-based probability of detection assessment and fusion approach for reliability evaluation of vibration-based diagnosis systems. *Struct. Health Monit.* **2020**, *19*, 649–660. [CrossRef]
41. Falcetelli, F.; Yue, N.; Di Sante, R.; Zarouchas, D. Probability of detection, localization, and sizing: The evolution of reliability metrics in Structural Health Monitoring. *Struct. Health Monit.* **2022**, *21*, 2990–3017. [CrossRef]
42. Kompanets, A.; Leonetti, D.; Duits, R.; Maljaars, J.; Snijder, H. Probability of detection curve for the automatic visual inspection of steel bridges. *Ce Pap.* **2023**, *6*, 814–824. [CrossRef]
43. Nondestructive evaluation system reliability assessment. *Dep. Def. Handb.* **2009**, *7*, 15.
44. Virkkunen, I.; Koskinen, T.; Papula, S.; Sarikka, T.; Hänninen, H. Comparison of â versus a and hit/miss POD-estimation methods: A European viewpoint. *J. Nondestruct. Eval.* **2019**, *38*, 1–13. [CrossRef]
45. Annis, C. R Package mh1823 (Version 7.3.4). Windows. 2023. Available online: https://statistical-engineering.com/mh1823/ (accessed on 22 May 2024)
46. *ASTM-E3023-15*; Standard Practice for Probability of Detection Analysis for â Versus a Data. ASTM International: West Conshohocken, PA, USA, 2015.

DOI: 10.3390/automation5030019
URN: urn:nbn:de:hbz:465-20250127-134422-6