Research

On using machine learning algorithms for motorcycle collision detection

Philipp Rodegast[1] · Steffen Maier[1] · Jonas Kneifl[1] · Jörg Fehr[1]

Abstract

Globally, motorcycles attract vast and varied users. However, since the rate of severe injury and fatality in motorcycle accidents far exceeds that of passenger car accidents, efforts have been directed towards increasing passive safety systems. Impact simulations show that the risk of severe injury or death in the event of a motorcycle-to-car impact can be greatly reduced if the motorcycle is equipped with passive safety measures such as airbags and seat belts. For the passive safety systems to be activated, a collision must be detected within milliseconds for a wide variety of impact configurations, but under no circumstances may it be falsely triggered. For the challenge of reliably detecting impending collisions, this paper presents an investigation towards the applicability of machine learning algorithms. First, a series of simulations of accidents and driving operation is introduced to collect data to train machine learning classification models. Their performance is henceforth assessed and compared via multiple representative and application-oriented criteria.

Article Highlights

- The challenge of reliably detecting a motorcycle collision within a short time by means of machine learning classification is investigated.
- Different machine learning architectures are compared in terms of their practical capability with metrics specifically adapted to the problem.
- Performance is validated on standardized ISO 13232 accident configurations.

1 Motivation

Globally, powered two-wheelers, more commonly called motorcycles, attract vast and varied users [1]. Depending on the region, they enjoy popularity as (i) a compact means of individual transportation in increasingly congested urban traffic and tight parking conditions, (ii) cheap individual mobility due to a lower price, lower energy consumption, and lower maintenance costs compared to, e.g., passenger cars, and (iii) a leisure or exciting sports activity due to its unique handling experience with the feeling of freedom when riding. However, their poor passive safety poses an excessive risk in road traffic at considerable social costs.

Discover

The risk of suffering severe injuries or dying in an accident is more than twice as high for motorcyclists than for occupants of cars. Worldwide, about 375,000 drivers and passengers of two- or three-wheeled vehicles die each year, according to [2]. In 2021, 302 (2.2%) motorcyclists killed and 5230 (38.2%) motorcyclists seriously injured were recorded in 13,702 motorcycle accidents with personal injury in Germany alone [3]. In that same period, in 160,771 car accidents involving injuries to persons, only 1433 (0.9%) were killed, and 30,902 (19.2%) severely injured car drivers were registered. The higher risk of injury and fatality is attributed to the lack of passive safety systems. While passenger cars benefit from extensive safety measures such as airbags and seatbelts in an enclosed passenger cell with an energy-absorbing crumple zone, motorcycles lack these features. There are many different active safety systems that improve the active safety of motorcycles [4] (including motorcycle-detecting automatic emergency braking which is integrated into passenger vehicles cars [5]). Studies show that they can potentially prevent a significant proportion of accidents, but they are not able to avoid all collisions in the future [5, 6]. Instead, nowadays motorcyclists rely on personal protective equipment such as helmets, safety clothing, and protectors, which, as the above statistics of accident outcomes show, have poor protective performance.

As part of a research project, the Institute of Engineering and Computational Mechanics (ITM) of the University of Stuttgart is investigating a new passive safety concept for motorcycles. The concept includes, among other features, seatbelts that restrain the rider to the vehicle and airbags in order to decelerate the rider in a controlled manner. The strategy for an impact scenario is no longer to detach the rider from the motorcycle so that, in the best-case scenario, to fly over the accident opponent without violent contact but to restrain and decelerate the rider with airbags and belts. Crash simulations of a set of frequent impact configurations show that the likelihood of injuries is significantly reduced by applying the novel safety concept [7]. Like other passive safety systems for PTWs (e.g., airbags for scooters from [8]), the concept depends on reliably detecting a hazardous impact opponent with the need for a sufficiently short decisional delay.

Available since 2007 with a front airbag, the Honda Goldwing GL1800 heavy touring motorcycle is the only motorcycle available on the market that detects an impact with an accident opponent to activate passive safety systems. However, because it is designed for a frontal impact, the sensor system in the front fork does not detect side or rear impacts [9, 10]. In [11], a whole range of other sensors are proposed to detect a motorcycle impact as soon as possible. A study on the proposed safety concept predicts that a large number of such sensors is needed to detect many accident configurations reliably without unwanted faulty activation in expected driving conditions [12]. As a result, the greater the number of sensors, the more complex and difficult it is to optimize and validate the decision-making process for activation. Ultimately, this is a very complex task for conventional methods such as threshold-driven detection logic and has strict requirements on validation and verification, see, e.g., [13]. Hence, this work is dedicated to the search for other methods capable of reliably detecting accidents within milliseconds. Algorithms from the field of machine learning are promising candidates for this task as, on the one hand, they are able to detect non-obvious patterns in data, which are therefore ignored in classical detection logic. On the other hand, many of them are very efficient and can therefore be quickly evaluated and used on hardware with limited computing power, such as found in the control units of a motorcycle.

Utilization of machine learning methods for crash predictions is already being investigated in a wide range of applications. Among others, many publications deal with automated real-time traffic supervision, which aims towards identifying potentially hazardous road sections. In [14], different deep learning and machine learning techniques are compared to predict real-time crash occurrence based on traffic data as well as weather information. Similar efforts are taken for crash detection as well as risk estimation [15] and identification [16] on freeways which could be used for traffic management. When it comes to crash predictions for motorcycles, on the contrary, the number of research projects and publications is lower. One field of research where machine learning (ML) methods are useful is the investigation on crash- or injury-severity in motorcycle accidents, which gives useful insights into how motorcycle riding can be rendered less dangerous [17]. Another general recurring approach for cars is to use smartphone-based crash detection, where the smartphone serves as a measurement tool as well as an automated emergency notification system [18–21]. Instead of using smartphone-based measurements, [22] use an ensemble-based crash detection on data emerging from dashboard cameras.

However, the possibility of using the motorcycle sensors themselves is often overlooked since most publications do not consider safety frameworks for motorcycles. Consequently, decisions are based on a few non-specialized measurements so that the detection performance is highly limited, also allowing for false positive detection. Furthermore, most publications rely on difficult-to-access and poorly available real-world data where it is hard to reproduce exactly those scenarios crucial for accident detection. As we use simulation models to generate the training data, we are able to generate specific and well-defined parametric scenarios describing a wide variety of typical accidents and non-critical driving scenes. To the authors' best knowledge, no research currently investigates the applicability of machine learning methods as a real-time, onboard crash prediction mechanism in order to trigger a given passive safety protocol.

   This leads to the fundamental research question of this work: Is a ML model capable of accurately identifying an impact in due time? Responding to this question, this paper deals first with producing an expressive database that is consequently used to train a selection of ML models and to evaluate their performance. The following describes the used simulation model, illustrates how the database is produced and trains a selected set of ML classification models on this data.

2 Materials and methods

First, the simulation model is described, which is used to collect both operational driving data (regular use) and collision data (accidents). The collision data exclusively includes impacts into a car as a collision partner. It does not include solo accidents, since this study merely serves as a proof of concept and the simulation model is reused from a preceding study [7]. Subsequently, data generation and processing are addressed. In the last section, ML methods are implemented and optimized.

2.1 Simulation model

Data collection is accomplished solely by means of simulation as it offers the opportunity to cheaply evaluate and measure various operational driving and collision scenarios. In addition, the motorcycle concept currently only exists virtually, and real-world driving and crash tests are costly and pose great challenges to represent targeted scenarios. The simulation model is implemented in Siemens Simcenter's Madymo (version MADYMO 2020.1), which serves as a multibody (MB) finite-element (FE) crash-simulation environment. Madymo is a simulation environment for physical systems with a focus on vehicle collision dynamics and passenger safety and injury assessment. It combines MB system capabilities for large rigid body motions as well as FE analysis for structural behavior. Here, simulating crash dynamics not only saves costs but can also serve to reduce time to market significantly.

2.1.1 Motorcycle model and accident opponent

The motorcycle is modeled with three masses, see Fig. 1, defined by their mass m, rotational inertia I, and geometry. The bodies are linked via kinematic joints. The motorcycle chassis is attached to the suspended front fork (FF) and rear swing (RS). The telescopic front fork moves linearly (s_{FF}), whereas the rear swing rotates (ς_{FF}). Both suspensions attach to their wheels (FW, RW) via hubs, around which these rotate (φ_{FW}, φ_{RW}). To model structural deformation in an impact, the front fork allows for angular deformation τ_{FF}. It depicts a rotation of the front fork inwards around the steering hub. As an impact opponent vehicle, the simulation setup comprises a collision partner, which is represented by a 1987 Ford Scorpio configured according to [23]. The model is part of a broader modeling and simulation strategy ranging from a multibody-system, a coupled multibody- and finite-element setup, to a full finite element model. The multibody model used here is chosen in order to simulate long scenarios that span many seconds. For a thorough overview of the models that also include the passive safety systems, be referred to [24]. It should also be noted, that in the scope of this research, the motorcycle model represents a category L3 vehicle according to the United Nations Economic Commission for Europe (UNECE) with an engine

Fig. 1 Three mass rigid body model of the motorcycle: A frame (Moto) and two wheels (FW, RW)
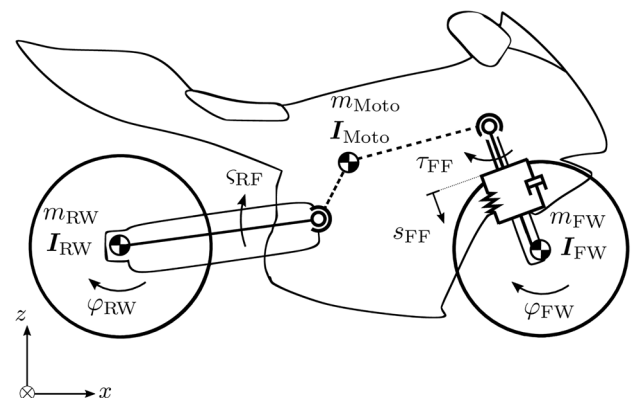
displacement larger than 50 m^3 or a maximum design speed of more than 50 km/h, since the referenced ISO 13232 accident configurations are restricted to L3 vehicles. However, the concept of this work can be applied just as well to L1 vehicles.

In contrast to investigations of passive safety measures in the course of an accident, which are already well-established, the period of interest for this study includes only a few moments after the impact. This means that the period of interest for this study occurs well before the passenger comes in contact with the airbags or is significantly restrained by the belts allowing to reduce the overall computational costs significantly. On the one hand, this allows stopping the crash simulations before any computationally complex passenger-safety-system interactions occur. On the other hand, the passive safety system, i.e., thigh belts and airbags, and the rider model can be excluded from the simulation setup as they have no qualitative influence on the simulation results in the period under consideration. These simulation setup modifications are beneficial for the overall numeric costs and allow for the execution of a far greater number of simulations than what would have been feasible within the same time with the inclusion of rider and passive safety systems.

A crucial issue with generating data from the model is that, due to complexity, the model does not allow for the application of lateral dynamics. Cornering behavior can thus not be incorporated into the dataset. Hence, only longitudinal dynamics are covered by the training data. This circumstance has far-reaching ramifications: Since, for example, the steering angle moves out of the 0° position in case of a head-on crash and by design never leaves this position during normal (non-crash) riding, the classifier would be induced false knowledge and would most likely decide solely on the basis of the steering angle signal. To circumvent this contingency, signals that contain information about lateral motion are to be strictly neglected in order not to overestimate the decision-making ability of a classification model. This poses a major restriction on parts of the available sensor data, as certain DOFs are only changed when the motorcycle collides with an opponent. Signals that are affected by the restriction are the motorcycle body's velocity and acceleration, both linear and angular.

In Table 1, all modelled sensors and available signals are listed. The first five signals are subject to dimensional limitation. The used component of these signals is given in the last column. In order to emulate the response of a tire pressure sensor, the resulting contact force between the crash opponent and each tire is combined with the contact force between each tire and the road surface, yielding the residual contact forces f_{FW} and f_{RW} for the front and respectively the rear wheel.

2.2 Data acquisition

A notable benefit of the proposed method is that the data used to train ML models does not come from logged real-world sensor data but from closely monitored simulations. This means that each individual sample can be assigned to a scenario and, therefore, also state (non-crash/crash). The knowledge about the state introduces the ability to use supervised learning methods. A switch is incorporated into the model in order to automate the labeling process. It is flipped as soon as a part of the motorcycle comes in contact with the car. The switch is allowed only one initial flip since, for the purpose of this elaboration, a crash does not stop until the simulation terminates. The switch reliably distinguishes between normal non-crash operation and crash scenarios, and is, thus, eligible to be further used as a class label for the machine learning classification application.

A fundamental operation of this investigation is to produce data covering the whole spectrum of both non-crash driving and crash scenarios. Training data must contain all necessary information to reliably differentiate between those two states, but it can only be composed of a multitude of individual simulations. The fact that the labeling process is not a task to be carried out manually but rather by the simulation itself enables to consider all necessary scenarios. Henceforth, a parameter-based simulation definition is used to fully exploit the fact that individual simulation postprocessing is not required. Consequently, a set of scenario parameters with an allocated range defines each subset. The individual subsets are then merged to form a comprehensive database.

In order for the parameter space to be sufficiently covered by a given amount of instances, latin-hypercube sampling (LHS) is applied. LHS, in contrast to simple-random-sampling (SRS), subdivides each parameter's range into equal-sized subdivisions, thus effectively partitioning the entire parameter space into hypercubes, called *strata*. A random sample is generated from each *stratum*, thus ensuring that the resulting distribution is fully representative of a given population [25]. All resulting simulative measurements are available under [26].

Table 1 Recorded output signals from the motorcycle simulation model

Abbreviation	Description	Unit	Component
Body lin. vel	Motorcycle body linear velocity	m/s	x
Body lin. acc	Motorcycle body linear acceleration	m/s^2	x
Body ang. vel	Motorcycle body angular velocity	rad/s	y
Body ang. acc	Motorcycle body angular acceleration	rad/s^2	y
FW ang. vel	Front wheel angular velocity	rad/s	
FW ang. acc	Front wheel angular acceleration	rad/s^2	
RW ang. vel	Rear wheel angular velocity	rad/s	
RW ang. acc	Rear wheel angular acceleration	rad/s^2	
RS ang. pos	Rear swing angular position	rad	
RS ang. vel	Rear swing angular velocity	rad/s	
RS ang. acc	Rear swing angular acceleration	rad/s^2	
FS lin. pos	Front suspension linear position	m	
FS lin. vel	Front suspension linear velocity	m/s	
FS lin. acc	Front suspension linear acceleration	m/s^2	
FD ang. pos	Front deflection angular position	rad	
FF ang. vel	Front-fork deflection angular velocity	rad/s	
FF ang. acc	Front-fork deflection angular acceleration	rad/s^2	
FW cnt. force	Front wheel contact force	N	
RW cnt. force	Rear wheel contact force	N	
FW/RW vel. diff	Wheel speed differential	rad/s	
FW/RW acc. diff	Wheel acceleration differential	rad/s^2	
cnt. sensor left	Contact sensor left	Bolean	
cnt. sensor right	Contact sensor right	Boolean	

2.2.1 Class *A*: uncritical data

Within the class of uncritical data, simulations are divided into two subsets. The first of which uses a sinusoidal base structure as a road profile. The second subset supplementary adds specified use cases to the database that can not be reproduced by the sinusoidal subset.

2.3 Scenario set *A.1*

A mesh of rigid shell elements which is created before each simulation serves as contact surface for the motorcycle. This road model has a total length of 300 m and a segment length of 0.2 m. The sine amplitude ranges from 0 m to 6 m. The first subset's road profile is described by a sinusoidal base profile with added noise in order to mimic road-unevenness. By ensuring that the interval length corresponds to the road length, the steepest section is limited to a road gradient of 12% which amounts to the maximum gradient found on open roads according to [27]. Phase shift and noise amplitude are also incorporated as parameters for additional variation. A set of 20 exemplary road profiles is depicted in Fig. 2. The figure additionally presents an expanded view in order to illustrate the superimposed noise. The motorcycle's initial velocity ranges from 3 m/s up to 23 m/s. The speed range is chosen to correspond to that of the crash scenarios (Fig. 6). Otherwise, if speed distribution in one class far exceeds that of the other class, the classification model would tend to incorporate that bias. Lastly, brake and acceleration torque acting on the wheel hubs are derived from one single parameter (since they cannot occur at once). In [28] it is shown that the mean braking torque of a motorcycle can be assumed to be 500 Nm and it is distributed, so that 70% is directed to the front wheel and 30% to the rear wheel. Likewise, [29] investigate the acceleration of high-performance race motorcycles. The authors state that the braking torque at the wheel can reach up to 1000 Nm. As a conservative estimate the acceleration torque limit is set to 500 Nm since the motorcycle under investigation here is not designed for racing. Scenario set *A.1* is composed of 100 entities that are parametrized via LHS. The parameters with their corresponding variation range are listed in Tab. 2.

Fig. 2 Road profiles for non-crash scenario set *A.1*. The road consists of a start platform which is connected to a randomly generated sinusoidal profile with a maximum gradient of up to 12% and a superimposed noise
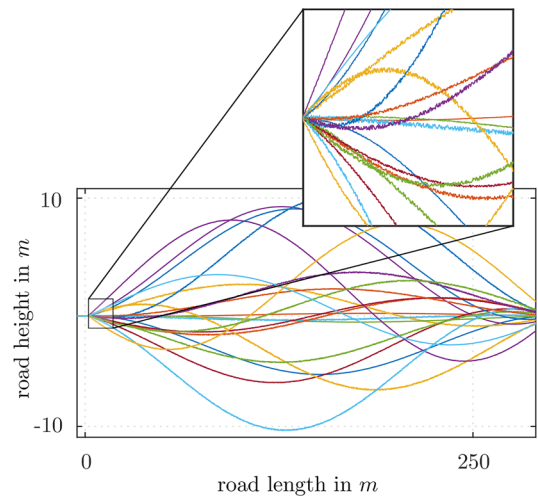

Table 2 Parameter space for non-crash scenario set *A.1*

Parameter	Range			Unit
sine amplitude	0	:	6	m
phase shift	0	:	360	°
road 'noise'	0	:	0.025	m
$v_{\text{Moto,init,A1}}$	−1.2	:	1.2	m/s
$t_{\text{accel/decel,A1}}$	-500	:	500	N m

2.4 Scenario set *A.2*

Although the parametrized scenario generation already covers a wide range of applications, some use cases can not be emulated by this method and should still be considered in the database. These scenarios, referred to as set *A.2*, are designed manually and appended onto the dataset. Namely, these simulations are intended to replicate the following use cases: (i) riding over potholes, (ii) approaching a curbstone, (iii) riding down multiple curbstones, and (iv) riding over speedbumps. Road profiles that are designed to imitate these obstacles are depicted in Fig. 3. In combination with initial velocities lying within the same range as in subset *A.1* , see Tab. 2 and using LHS as sampling method, a total of 20 simulations (4 in each use case) are performed to build subset *A.2*.

2.4.1 Class *B*: crash data

The data containing accidents involves more than just head-on collisions, where simple threshold-based decision logic would suffice for detection. Instead, the aim is to examine a broad range of conceivable impact scenarios. Consequently, parametrized scenario generation is resorted to again. A Dutch study on motorcycle accidents shows, that almost 70% of the recorded accidents involved an impact against an opposing vehicle [?]. The study further shows, that no one accident configuration with an opposing vehicle is overly dominant, but rather, that a multitude of accident scenarios with equal probability exist. These include a variety of intersection collisions, where either the car or the motorcycle are making a left-hand turn. However, overtaking and making a U-turn are also likely scenarios. This leads to the conclusion that it is not sufficient to limit the analysis to one type of collision, but that many possible scenarios should be considered. Two different base architectures are therefore designed to cover the spectrum collisions. A stationary car is struck by a motorcycle at different angles and velocities (subset *B.1*) and a car striking the stationary motorcycle at different angles and velocities (subset *B.2*). Additionally, a set of ISO 13232 crash scenarios are simulated and preserved in order to validate the classifier's ability to generalize after being trained.
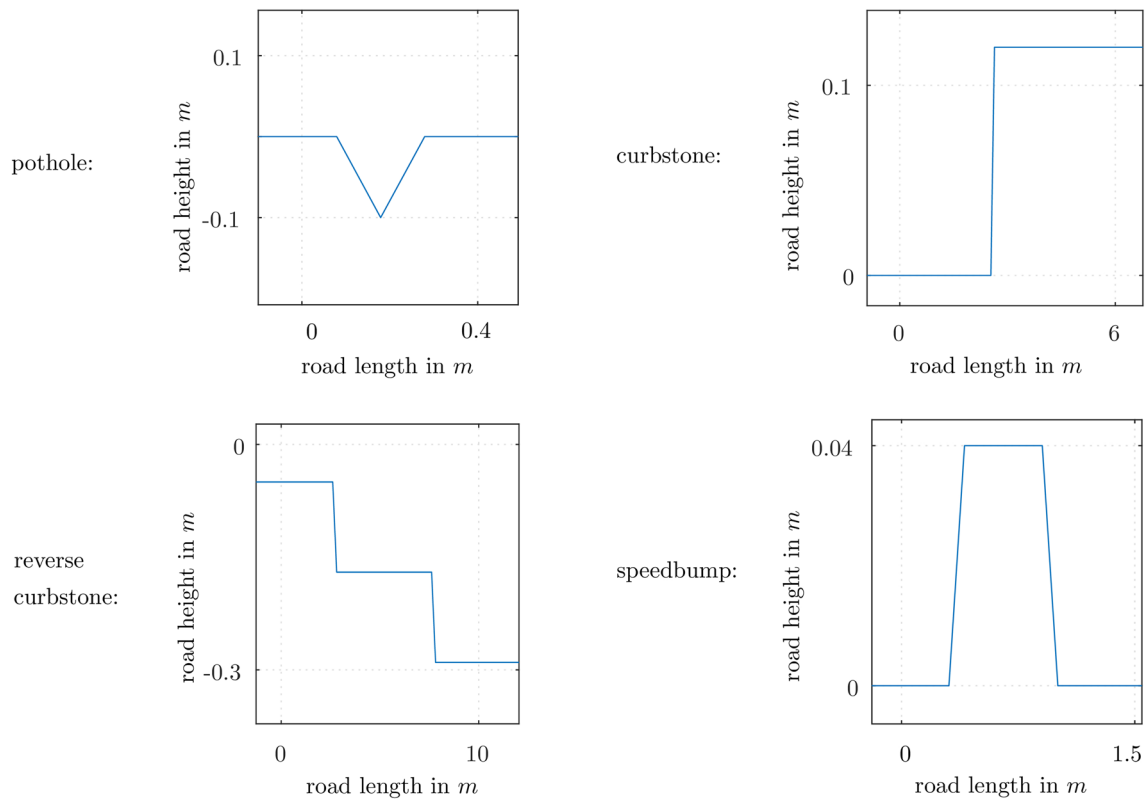
Fig. 3 Additional road profiles for non-crash scenario set *A.2*

2.5 Scenario set *B.1*

Set *B.1* consists of 100 simulations in which the motorcycle with an initial velocity $v_{\text{Moto,init,B1}}$ approaches a stationary car that is both rotationally (α_{B1}) and laterally ($o_{y,\text{B1}}$) displaced. Figure 4 illustrates the structure of the scenario set *B.1*. By offsetting the car by y_{offset}, grazing-accidents are included in the scenario database. Table 3 shows the range in which the parameters are varied.

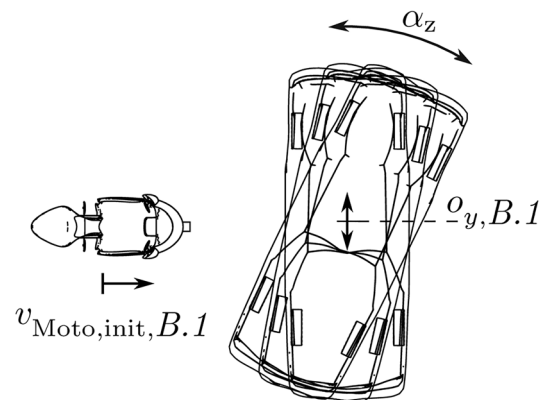**Fig. 4** Schematic illustration of crash scenario set *B.1*

Table 3 Parameter space for crash scenario set *B.1*

Parameter	Range			Unit
$v_{\mathrm{Moto,init},B.1}$	6.7	:	13.4	m/s
α_z	0	:	360	°
$o_{y,B.1}$	−1.2	:	1.2	m

2.6 Scenario set *B.2*

For more variation, in *B.2* the setup is inverted with the motorcycle now stationary and the car pointing at the motorcycle at different angles, see Fig. 5. Additionally, the direction of rotation of the car is shifted by γ_{B2} for further variation of the point of impact. Table 4 lists the range of variation for each parameter assigned to set *B.2*.

2.7 Validation set *B.3*

Validation scenarios are implemented in order to evaluate the final performance of the trained models and, thus, their ability to generalize. Scenarios according to [30] are selected for this purpose in order to be representative. The ISO norm 13232 provides a set of impact configurations based on a statistical analysis of real-world crash events. The full set consists of 25 impact configurations, shown in Fig. 6.

2.8 Machine learning classification

The remainder of the chapter is devoted to the task of building machine learning models using the `scikit-learn` toolbox in Python [31]. The section is structured into preprocessing of the raw data, model preselection, and hyperparameter-tuning using grid-search in combination with cross-validation (CV).

2.8.1 Preprocessing

Data preprocessing can often lead to a significant performance enhancement compared to working with raw data. Preprocessing includes distribution management, standardization, checking for, and handling of missing values.

As some classification algorithms tend to perform better or even depend upon standardization, a scaling routine is implemented prior to the classification. Whereas some classification algorithms like decision trees and ensembles do generally not require data standardization, others, like, for example, neural networks, are strongly reliant on it. Otherwise, features that have different magnitudes are not treated equally. Standardization scales each feature to unit variance. In

Fig. 5 Schematic illustration of crash scenario set *B.2*
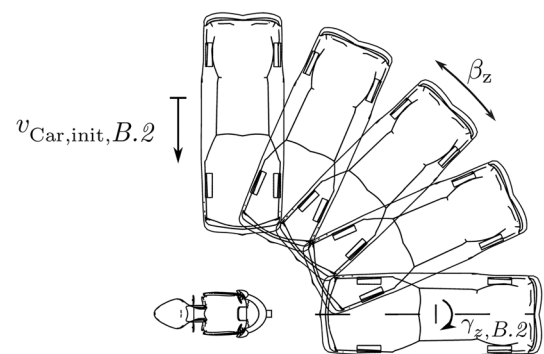
Table 4 Parameter space for crash scenario set *B.2*

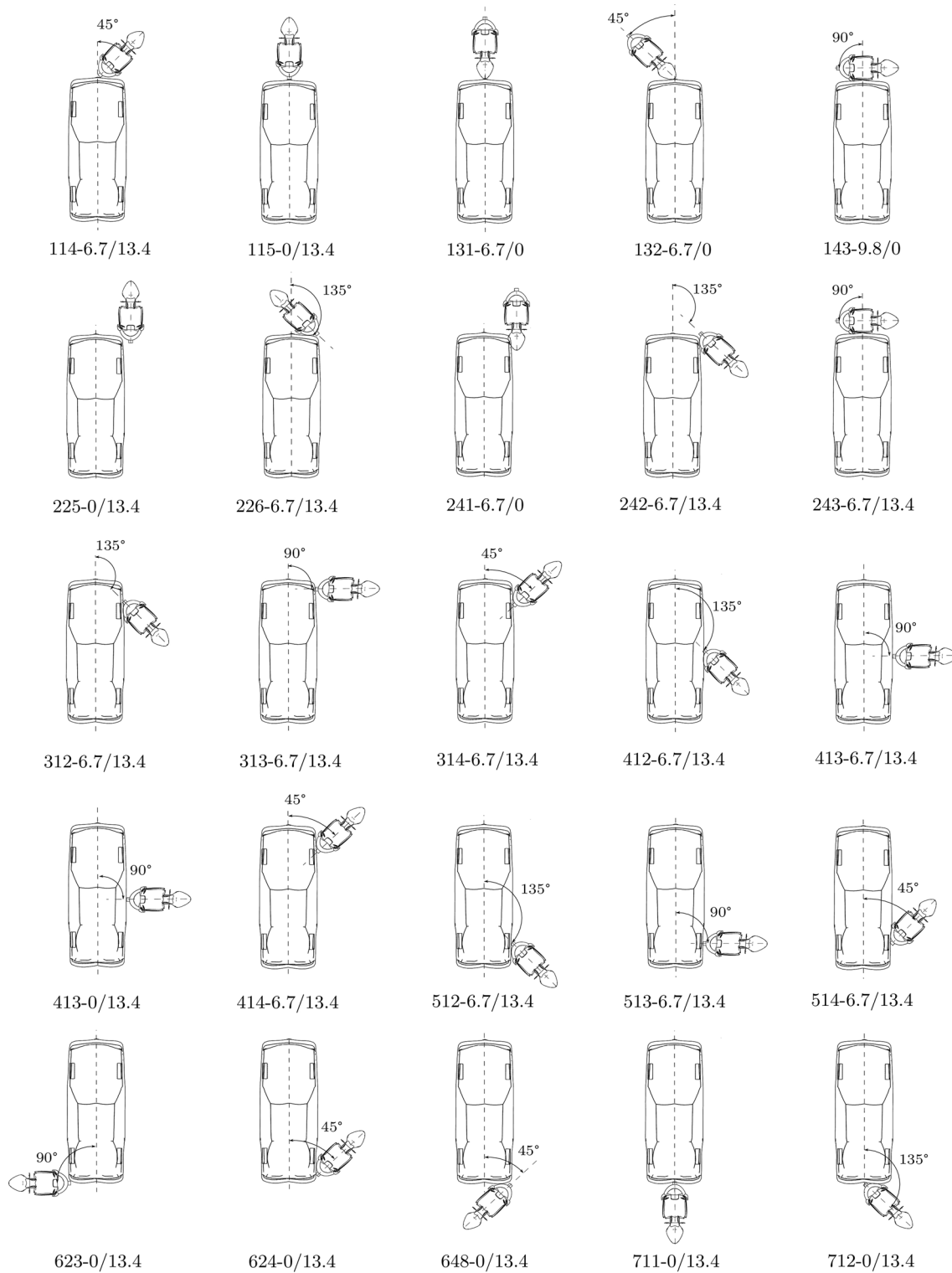| Parameter	Range			Unit
$v_{\mathrm{Car,init},B.2}$	6.7	:	13.4	m/s
β_z	0	:	360	°
$\gamma_{z,B.2}$	-5	:	5	°

Fig. 6 Scenarios of the validation set *B.3* from [30]. Each scenario is described by a code XXX-Y/Z, where XXX encodes the relative position of car and motorcycle, Y is the car's, and Z is the motorcycle's impact velocity in m/s

order to be consistent, a transformation vector is computed on the training set and applied to the test set rather than scaling each dataset to unit variance in its own right. The standardized value $x'_{i,m}$ for each sample $x_{i,m}$ of timestep i and feature m is carried out via

$$x'_{i,m} = \frac{x_{i,m} - \bar{x}_m}{\sigma_{x,m}}$$

with $\bar{x}_m$ being the arithmetic mean of feature $\boldsymbol{x}_m$ and $\sigma_{x,m}$ its standard deviation.

An additional preprocessing method is feature extraction via principal-component-analysis (PCA). It aims at deriving meaningful and non-redundant variables from the original dataset by projecting it to a lower-dimensional space by means of singular value decomposition (SVD). By doing so the original dimension of the feature space dimension, which is with a total of 23 features fairly high, could be reduced, This could be beneficial considering that the concluding model is to be run in real-time on an embedded system. However, the desired effect does not materialize well and the method proves to be ineffective for this application. Consequently, it will not be addressed further in this report.

2.8.2 Training-test-split

A decisive circumstance to consider when subdividing time-dependent data is that neighboring samples have a tendency to be located in close proximity to each other. A random training test split, like it is often performed on non-time-dependent data, is, henceforth, not an appropriate splitting method.

In the case of this application, training-test-split is carried out on whole coherent simulations itself rather than on samples. As this is a safety-critical application, a 50-50 split is performed, meaning half of the available data is retained for testing to cover a broad range of scenarios. The remaining half is dedicated towards training the models. Additionally, none of the ISO scenarios of set *B.3* shall be included in the training set as it is of particular interest to assess how well-trained models generalize on these representative scenarios.

When training a model for classification purposes, close attention must be paid towards the distribution of the two classes. Since crash-labeled samples are, in this case, much less frequently represented in the raw dataset, non-crash-labeled data requires subsampling in order to achieve equal distribution. The desired distribution is achieved with a sampling rate of twelve. The resulting sizes and distributions of the two datasets are displayed in Tab. 5.

For the final application of the real-time classification model into the virtual "system" of the motorcycle, a sample rate of 2kHz is selected. This is a sample rate that most commercially available sensors are able to safely handle and which also leaves a sufficient margin of samples for the implementation of an activation threshold. In addition to the above-mentioned training and test datasets, which are no longer subject to a uniform sample rate, individual scenario datasets are prepared, which incorporate the realistic scenarios from [30]. Those simulations are synchronized to the selected sample rate of 2kHz in order to evaluate the model's decisional delay.

2.8.3 Model preselection

The amount of available classification algorithms is extensive and thus not feasible to investigate exhaustively. A preselection of suitable algorithms is therefore shortlisted. The preselection is designed to incorporate multiple different classification approaches like ensembling, support vector machines (SVMs), and artificial neural networks (ANNs) as they are some of the most frequently used algorithms available. In Tab. 6 all preselected models and their underlying method are listed.

2.8.4 Hyperparameter-optimization

A decisive determinant for the performance of a classification model is the proper choice of its hyperparameters. There are several different approaches to resolve this problem, such as random or grid searches. In the scope of this paper,

Table 5 Distribution of samples dedicated towards training and testing

Dataset		Class	Samples
training	$\Big\{$	Non-crash	13, 842
		crash	10, 908
		total	24, 750
test	$\Big\{ \cdot$	Non-crash	11, 238
		crash	11, 778
		total	23, 016

Table 6 Preselection of scikit-learn classification algorithms

Algorithm	Base method
AdaBoost.M1	Ensembling (boosting)
Gradient boosting	Ensembling (boosting)
Random forest	Ensembling (bagging)
SVM	Support vectors
MLP	Feedforward neural network

grid search parametrization, in combination with cross-validation (CV), is employed in order to discover the best-fitting parameters for each model.

Using grid search optimization, the (discrete) hyperparameter space searched within must be predefined. A classification model is trained on each combination of parameters, and, using cross-validation, a mean score is computed for each iteration. The type of score as well as the number of cross-validation folds are user-defined. For this application 20-fold CV in combination with the F_1 score

$$F_1 = 2\ \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

is selected since the dataset consists itself of a multitude of subsets. The F_1 is chosen as it combines both precision and recall scores.

Each parameter's grid is incrementally refined, effectively narrowing the search radius until further refinement no longer yields an improvement. Tuned resulting hyperparameters for all five models are listed in Tab. 7. Only parameters that deviate from the standard parameters are listed.

2.8.5 Baseline model

In order to put the performance of the classification models into perspective, a simple threshold based model acts as a baseline. The model is implemented according to [9], where a set of accelerometers is placed at the fork legs of the motorcycle. The signals are then used to determine if the motorcycle is about to collide. Since the two acceleration signals are averaged, in this paper there is only one accelerometer placed at the front hub. The models only parameter is the threshold at which it detects an impending collision. This threshold is tuned on the training data in the exact same way as for the other models, with its hyperparameter-space being only one dimensional.

Table 7 Tuned hyperparameters for the five classification methods

Algorithm		Hyperparameter	Value
AdaBoost	{	Number of learners	80
		max. leaner depth	2
		learning rate	0.6
		algorithm	SAMME
Gradient Boost	{	Number of learners	80
		max. leaner depth	10
		learning rate	0.5
		loss function	exponential
Random Forest	{	Number of learners	45
		max. leaner depth	17
		splitting criterion	entropy
SVM	{	C	120
		Gamma	auto
Neural Net	{	Number of hidden layers	1
		hidden layer size	220
		initial learning rate	0.002
		optimization tolerance	0.0003
		early stopping	enabled

3 Results and discussion

In this section, the results from the previously introduced trained classification models are compared to each other and are individually evaluated for fitness. ML classification-specific criteria on one side, as well as several application-specific criteria, act as a basis for comparison. Furthermore, available sensor data is examined and ranked according to its individual contribution. The computation of feature importance can be helpful to point out wether the feature space dimension can be reduced. The final section describes the individual feature contribution, giving an overview of sensor significance that can assist future work.

3.1 Capability assessment

The following sections serve as an illustration of the trained classification models' performance. This allows to draw a comparison between the models and assess the level of proficiency that can be expected from a specific model. In addition to a selected set of machine learning performance metrics, a few domain-specific criteria are established. The main objective is to outline the overall performance as broadly as possible in order to make a reasoned choice when selecting a model.

3.1.1 Performance measures

Performance metrics that are considered to evaluate and compare the achieved training results are the receiver operating characteristic (ROC) curve, which shows the true positive (TP)-false positive (FP) trade-off of a model. The area under the ROC curve (AUC) quantifies the individual trends. Additionally, scores that are computed from the confusion matrix are listed. For the sake of clarity, the resulting confusion matrix of all models is shown in the appendix (Fig. 1).

3.1.2 Receiver-operator characteristic

The resulting receiver operator characteristics for all trained models are displayed in Fig. 7. All curves lead through the sweet spot in the upper left corner, where classification yields a high TPR while maintaining a low FPR. The results indicate that there is no direct trade-off for all models between achieving a high TP rate (TPR) and keeping the FP rate (FPR) low. The ROC curves show an almost identical trend for four of the models, with only the AdaBoost model's performance being marginally poorer. The ROC curve does, therefore, not yet permit any statement about performance fluctuation between the models.

3.1.3 Machine learning scores

Since one single index is not able to sufficiently describe the performance of a classification model, a selection of metrics is chosen. The intention is not only to point out which models perform well and which are rendered unfit but also to indicate potential overfitting by computing the score on the training data. Consequently, a well-fitted model will tend to yield similar scores on both sets sacrificing training *accuracy* for better generalization ability. In contrast, an overfitted model tends to achieve a higher score on the training set than on the test set. The resulting scores are collected in Fig. 8.

Fig. 7 Receiver-operator characteristic of all five models. A perfect classification model would reside in the top-left corner, reaching a 100% TPR while maintaining a 0% FPR, essentially, mapping each sample flawlessly to its associated class
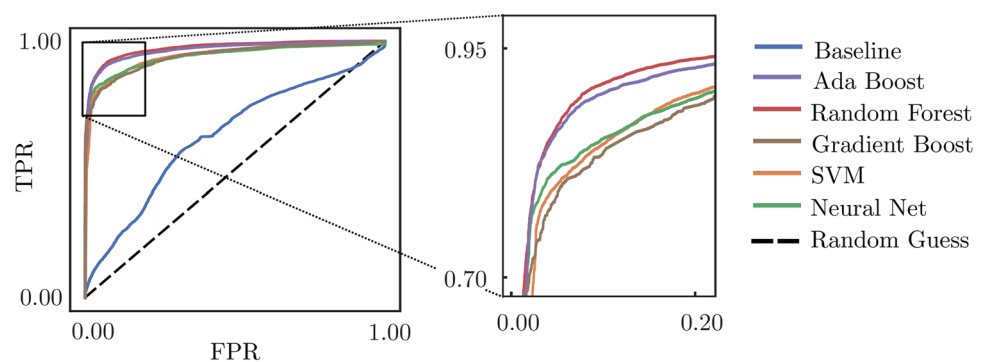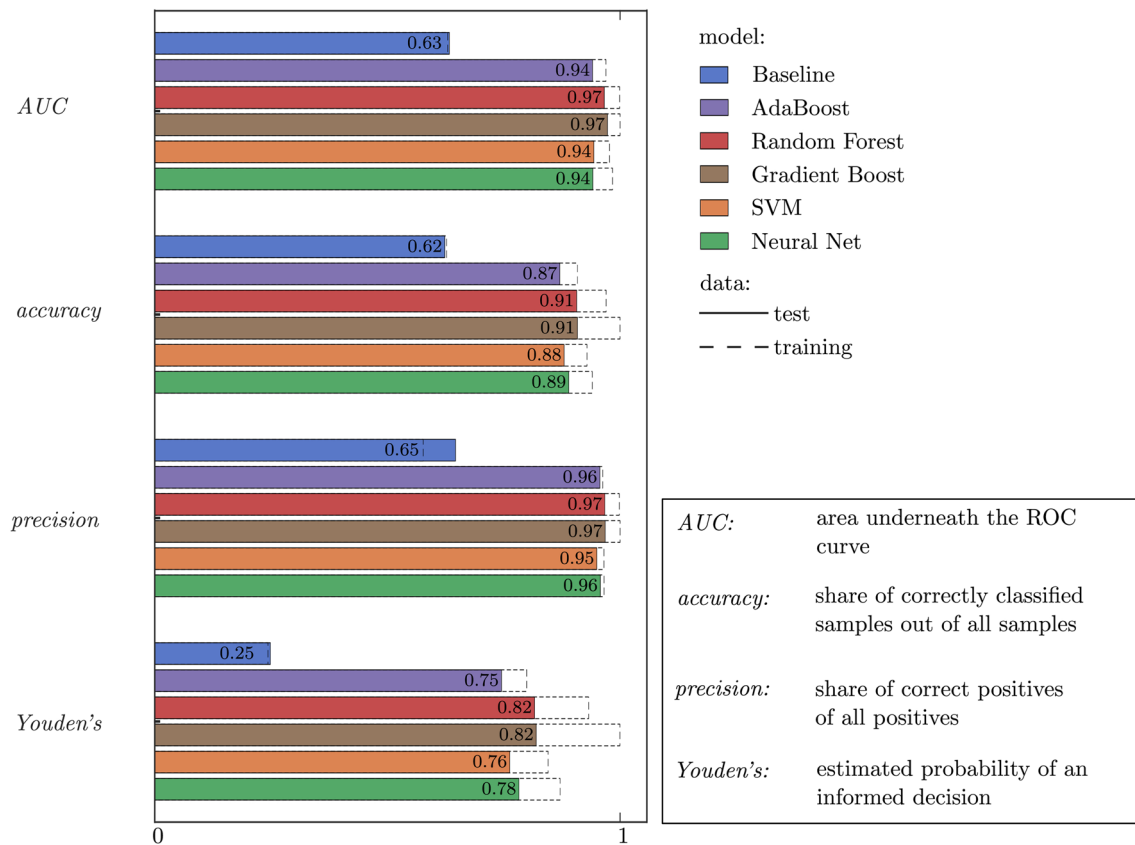
Fig. 8 Performance scores of all five trained models computed on both training and test data. A large deviation between training and test score is a strong indicator for and overfitted model

- The *AUC* Score quantifies the trend given by the receiver-operating characteristic, by calculating the area underneath the curve. A higher score means the demand for a better TPR does not tend to sacrifice a models FPR and vice versa.
- *Accuracy* measures the rate of correctly classified samples of both classes out of all samples. It is thus a valuable indicator of overfitting, if a model yields a sufficiently lower score on the test set than on the training set.
- The *precision* score, which in this context accounts for the classifier's ability to not falsely misclassify a non-crash sample as a crash. Sufficient scoring on this assessment is of elementary importance for applying the method presented in this report. Frequently misclassifying class *A* samples and, thus, falsely initiating deployment of passive safety mechanisms render the implementation of a ML-based crash detection algorithm possibly more harmful than profitable.
- As a combination of sensitivity and recall, *Youden's* index measures the overall informedness of a model's decision-making process.

Summarizing the scores and discrepancies presented in Fig. 8 permits to conclude the models' performance and generalizing ability. At first glance it is clearly visible, that the baseline model is not able to keep up with the ML models in terms of performance. Its scores are significantly lower than those of the other models. The poor scores indicate, that the model may be able to make a decision, albeit not an informed one.

   In comparison to the baseline model the ML models tend to achieve a similar performance, with the *AUC* scores only ranging from 0.94 to 0.97 and *precision* being almost identical. The *accuracy* and *Youden's* scores however hold more information. Firstly, the test scores of the Random Forest and Gradient Boost models are slightly higher than that of the other models. However, as the dashed line suggests, their ability to generalize tends to be impaired due to overfitting. To summarize, all of the five models achieve a similar performance with two of the models being subject to overfitting.

Table 8 Number of successive samples and equivalent time for a valid classification for each trained classification model

Model	$n_{activation}$	Activation time	Unit
Baseline	36	18.0	ms
AdaBoost	26	13.0	ms
Random Forest	19	9.5	ms
Gradient Boost	25	12.5	ms
SVM	16	8	ms
Neural Net	9	4.5	ms

3.1.4 Crash prediction requirements

Besides the aforementioned performance criteria which assess the classification models themselves, there are certain requirements that are given by the very nature of what is intended to follow the crash prediction algorithm. Theses requirements are that (i) no false detection is raised when not at risk for accidents, that (ii) prediction time is sufficiently short, for the airbag to deploy fully before the rider impacts the motorcycle, and that (iii) the model is computationally efficient. The first domain requirement results from the fact that the algorithm is intended to initiate passive safety precautions, such as the deployment of the airbag and fastening of the thigh belts. Some of the used airbags are non-deflating and, thus, stay inflated for at least a few seconds. False deployment should, therefore, be avoided at all costs due to the obstruction of visibility and maneuverability as well as rider shock and product reputation. Requirement (ii) aims to keep the prediction time of the classification models modest since the airbag inflation takes a comparably long time due to a much larger volume than that of a passenger car. In this context, a prediction time lower than 12 ms has proven to be sufficient in order for the airbag to inflate fully in time [11]. Furthermore, the fundamental idea behind this investigation is that the classification model operates in real-time on the motorcycle's embedded system. It is therefore reasonable to analyze and compare the latency of the classification models since CPU capacity can be assumed to be seriously limited. The requirements are tested on the ISO scenarios presented in Fig. 6 as well as a share of sets *A.1* and *A.2* that are preserved for the test set and that act as control scenarios. Thus, all data the classifier is tested on is not included in the training set and is not used for hyperparameter-tuning.

3.1.5 Decisional delay

For the purpose of meeting requirement (i) and eliminating false-positive detection of outliers, an activation threshold is introduced, which acts as a fail-safe mechanism. The activation threshold determines, how many consecutive positive classifications are needed so as to not make a false prediction in a safe scenario. The parameter is therefore tuned towards not making a *crash* prediction within the *non-crash* scenarios in the training data. Assuming then, the imaginary microcontroller running the algorithm has a fixed cycle time t_{MC}, a flawless classification model would have a prediction time of $(n_{activation} \cdot t_{MC})$.

    Requirements (i) and (ii) contradict each other, as can be concluded from a conceptual experiment concerning the ROC curve. If the activation threshold $n_{activation}$ is set arbitrarily high, the rate of false positives will diminish, however, prediction time would be infinite. In the opposing case, the activation threshold is set arbitrarily low. In this case, prediction time would be sufficiently short, but at the cost of having misclassifications. The link between requirements (i) and (ii) can be found in the prediction time. The schematic process is explained in Fig. 9. The threshold values are tuned individually for each model so that no false prediction is made in control scenarios. Classification models that are subject
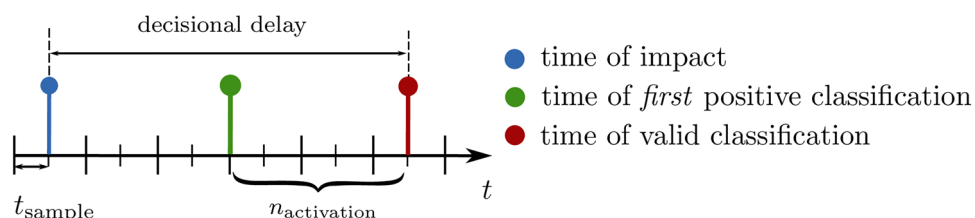

Fig. 9 An exemplary detection process, beginning at the time of impact and ending with a valid classification. A valid prediction requires a specified number of consecutive positive classifications in order to filter out any false positives

to a higher FP rate are consequently assigned a higher activation threshold $n_{activation}$. The individual activation thresholds are listed in Tab. 8.

From Tab. 8 it becomes apparent that at a sample rate f_{sample} of 2kHz and an activation threshold $n_{activation} = 26$ and 25, the AdaBoost and Gradient Boost models are rendered unfit as they could not meet the required decisional delay of 12 ms even with a perfect *accuracy* score. The set of ISO crash scenarios is categorized for the purpose of the decision delay assessment. Categories are *frontal* contact crashes, *lateral* or *rear* contact crashes, and lastly, *grazing* accidents. Decision delay is interpreted as the time between initial contact and the moment when $n_{activation}$ successive positive classifications are reached. The mean decision delay each model achieves in each of the three crash categories is presented in Fig. 10. The illustration presents, first of all, an intuitive trend regarding the three crash categories. The delay when predicting *frontal* accidents, on average, is significantly lower than that of *grazing* accidents, meaning they are effectively easier to predict. This can partly be attributed to the acceleration that the motorcycle exhibits during an accident, which is generally larger in magnitude in *frontal* crashes than during *grazing*. The figure shows that only the Neural Net model is capable of making an informed decision within the time limit for *frontal*, *lateral*, and *rear-end* accidents. The SVM model, though failing at detecting *frontal* accidents, predicts *lateral* and *rear-end* accidents in due time. The other models are not able to decide within the required period of time as their decisional threshold does not allow for as much agility. *Grazing* accidents pose a serious challenge for all of the models equally, as no model manages to stay within the time limit.

3.1.6 Runtime

To conclude the capability-assessment-investigation, the runtime of each model is evaluated. Runtime, in the context of real-time capability, is the time that elapses between the input of an instruction and the results output by the computing unit. It shall be stated that the results of this evaluation have no context outside this work and are merely a concept to compare the latency distribution of the six trained models. runtime measurement is carried out on an Intel® Core™i7-2600 CPU which operates at a base frequency of 3.4[]GHz. The runtime mean and standard deviation are computed for each model on the same test set and on the same processing unit.

The results are listed in Tab. 9 for each of the models. The distribution of the mean values shows that incremental sample classification for the Gradient Boost, SVM, and Neural Net model takes considerably less time than it takes for the remaining two models. With latency ranging from (0.05 to 0.2)ms these classifiers require only a fraction of the time of AdaBoost and Random Forest, where latency is distributed from (4.29 to 5.60)ms. It is to be expected that the runtime of the baseline model is small, as its computational cost is moderate. The low runtime of the models Gradient Boost,

Fig. 10 Prediction delay for the control scenarios [30] from Fig. 6 categorized into 3 accident groups
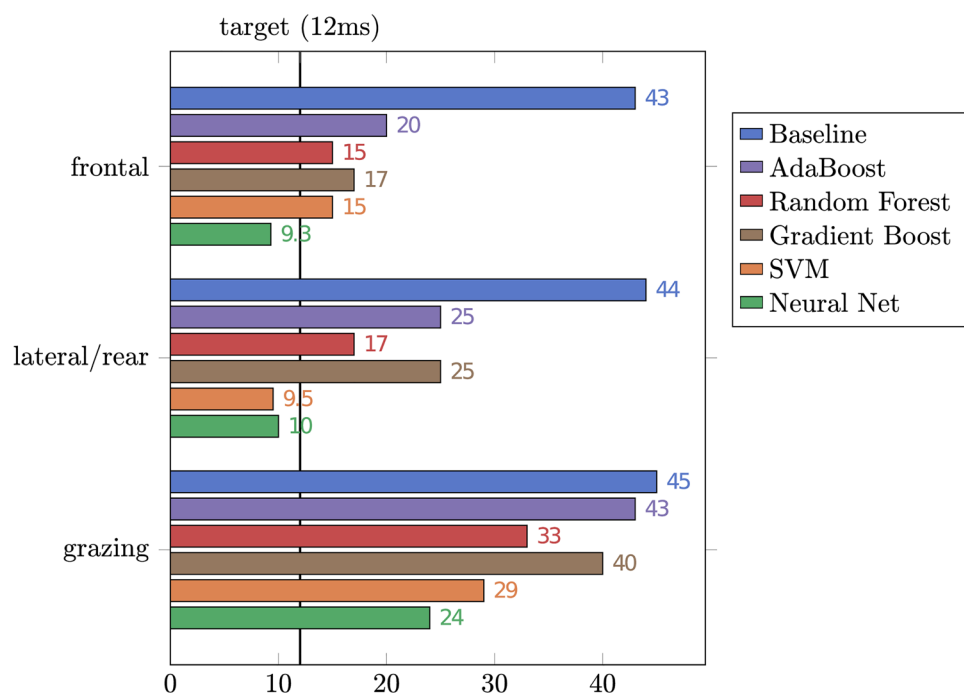
Table 9 Mean computation time of the different models and standard deviation carried out on an Intel® Core™i7-2600 CPU

Model	Runtime mean	std. dev.	Unit
Baseline	0.02	0.001	ms
AdaBoost	4.29	0.14	ms
Random Forest	5.60	2.27	ms
Gradient Boost	0.16	0.02	ms
SVM	0.20	0.004	ms
Neural Net	0.05	0.002	ms

SVM, and Neural Net compared to that of the remaining models indicate that they should be preferred especially when real-time capability is required.

3.2 Feature importance

In order to achieve the results shown above, not all features are of equal importance to the classification model. Some features are of greater use in order to differentiate between categories than others. From an economic perspective, it is particularly interesting which features contribute only marginally towards the overall result as it allows the sensors to be dispensed with while maintaining comparable performance. For the following investigation only the results of the Neural Net are used as it yields the best overall performance and meets the requirements. To find the most important features for ANNs, permutation-based feature importance is calculated by performing a number k of sample interchanges in individual features c and computes the resulting *accuracy* score of the corrupted dataset. For each iteration $k \in \{1, ..., n\}$ samples of column c are interchanged and the resulting score $s_{k,c}$ is evaluated. A final measure i_c for the importance of feature c is given by

$$i_c = s - \frac{1}{n} \sum_{k=1}^{n} s_{k,c},$$

Fig. 11 Permutation feature importance for the Neural Net model
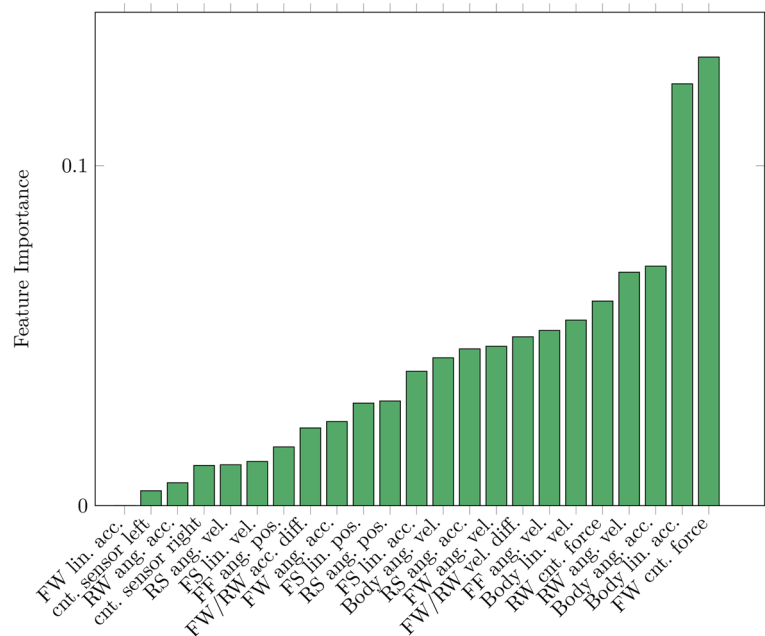
where *s* marks the score on the uncorrupted dataset. The contribution of individual features is presented in Fig. 11 in progressive order so that in the case that feature dimension is limited, the features to omit are highlighted.

The permutation feature importance ranks that of the motorcycle body's linear acceleration (*Body lin. acc.*) one of the highest, which intuitively appears reasonable. After all, a common condition that all crash scenarios share is an excessive (negative) acceleration of the system due to a more or less severe impact. However, as shown by [10], the frame acceleration sensor alone is not a sufficiently agile crash indicator as it responds with a considerable time delay. The delay is mostly due to the deformation that occurs before the frame is actually decelerated. Furthermore, the model assigns a critical weight to the approximated tire pressure (*FW cnt. force*), which is derived from the contact force. Front and rear wheel, as well as frame angular acceleration, are also among the top-ranked signals. Signals that only seem to contribute marginally are the suspensions positions and velocities, both front (*FS lin. pos.* and *FS lin. vel.*) and rear (*RS ang. pos.* and *RS ang. vel.*). The contact sensors that are attached to both sides (*cnt. sensor left/right*) of the frame also do not particularly enhance the performance.

Concluding the feature importance investigation, it can be established that acceleration signals generally are of greater benefit towards the overall decision-making quality. A fact that is particularly convenient, as accelerometers are among the most widely used sensors. In contrast, position and velocity signals do not contribute equally as much. This also serves as a potential explanation for the prolonged mean decisional delay that is observed in grazing accidents for all five models. Since in grazing accidents, the induced (negative) acceleration is not as pronounced as in other types of accidents, the acceleration-dominant classifiers fail to respond in due time. This forfeit, however, is admissible since it implies that severe accidents with more pronounced acceleration are reliably detected.

4 Summary and conclusion

The outline of this elaboration is to implement and evaluate machine learning methods to detect motorcycle collisions early and reliably. Simulation data is used for this purpose as it is easier and more cost-saving to acquire. A further benefit is that it allows close monitoring and, thus, enables the utilization of supervised learning. For this reason, the multi-body simulation model is adjusted to incorporate automated data labeling without requiring manual intervention. An extensive database is, henceforth, produced by means of parametrized scenario generation in order to broadly cover both areas of interest.

The simulation model has some limitations, e.g. it does not allow cornering. Thus, this work cannot illustrate all possible accident scenarios. Instead, this work suggests a workflow which must be extended in the future for a holistic consideration. The signal selection and data processing take into account the influences, e.g. the non-consideration of cornering by not inducing "false knowledge" into the database. The database is used to train a selected set of machine learning classification models. The mindfully preselected algorithms that are under consideration are three decision tree ensemble models, an support vector machine as well as an artificial neural network. In order to have a better understanding of the results, a primitive, threshold based model acts as a baseline for comparison.

Challenges that are imposed on the prediction model are that (i) it raises no false detection during normal operation, (ii) it responds sufficiently quick to account for the airbag's prolonged inflation time, and (iii) it is computationally efficient for realtime application. In order to evaluate each individual classification model's performance and to rate their fitness, a sophisticated scoring procedure is established in Sect. 3, allowing for a reasoned choice. Performance is evaluated on a set of frequent crash scenarios from ISO 13232 to be representative. The classification models are individually tuned towards not making a potentially harmful false prediction. Only one out of the five trained models is able to reliably identify the more severe crashes flawlessly and in due time, with only the less harmful grazing accidents posing a challenge. Despite being fail-safe during non-crash driving operation, this model predicts crashes with sufficient agility

for the airbag to inflate within the prescribed time. At the same time the neural net model proves to be amongst the lowest in terms of computation time.

Although the scores of other models were superior, the Neural Net model still prevails, as it provides the most reliable classification and thus requires the lowest activation threshold of all the models. Furthermore, this work gives insight into the significance of individual sensor data via feature importance assessment. This examination reveals that, in general, acceleration signals are weighted much more favorably than position and velocity signals. This circumstance renders grazing accidents particularly difficult to predict, as the acceleration imposed on the motorcycle is considerably lower than in other collision types.

Despite the exciting contributions of this study in detecting motorcycle collisions for a variety of simulated scenarios, it is important to acknowledge its limitations, particularly with regards to the restricted computational capacity of in-vehicle control units and the lack of consideration of lateral dynamics due to simulation model shortcomings. While realistic boundary conditions are adopted, wherever feasible (e.g. sensor sample time, ISO crash scenarios), it should be noted that, for the sake of simplification, ideal measurement hardware is assumed. Any loss in signal quality, that any commercially available sensor would impose is neglected, and must be taken into account in future studies. Therefore, in subsequent research, it is of interest to first consider even more realistic and extensive scenarios and mimic real sensor measurements by adding noise to the data or using sensor models. In a second step, it is of interest to investigate the suitability of the prediction algorithms for real-time use on weak hardware.

Data availability The datasets generated for this study can be found in the DaRUS repository [26]. The dataset provides time histories of sensor data in *.csv format, classified into "accident" and "no accident".

Declarations

Conflict of interest The authors declare no Conflict of interest.

Appendix

See Fig. 12.

Fig. 12 Confusion matrix of all classification models evaluated on both training and test data

True Class		Model	Predicted Class			
			Non-Crash		Crash	
			Train	Test	Train	Test
Non-Crash		Baseline	9232 / 66.70%	7483 / 66.59%	4610 / 33.30%	3755 / 33.41%
		AdaBoost	13494 / 97.49%	10820 / 96.28%	348 / 2.51%	418 / 3.72%
		Random Forest	13825 / 99.88%	10899 / 96.98%	17 / 0.12%	339 / 3.02%
		Gradient Boost	13842 / 100.00%	10906 / 97.05%	0 / 0.00%	332 / 2.95%
		SVM	13502 / 97.54%	10738 / 95.55%	340 / 2.46%	500 / 4.45%
		Neural Net	13493 / 97.48%	10811 / 96.2%	349 / 2.52%	427 / 3.80%
Crash		Baseline	4620 / 42.35%	4916 / 41.74%	6288 / 57.65%	6862 / 58.26%
		AdaBoost	1915 / 17.56%	2564 / 21.77%	8993 / 82.44%	9214 / 78.23%
		Random Forest	723 / 6.63%	1810 / 15.37%	10185 / 93.37%	9968 / 84.63%
		Gradient Boost	4 / 0.04%	1776 / 15.08%	10904 / 99.96%	10002 / 84.92%
		SVM	1419 / 13.01%	2269 / 19.26%	9489 / 86.99%	9509 / 80.74%
		Neural Net	1131 / 10.37%	2114 / 17.95%	9777 / 89.63%	9664 / 82.05%

References

1. Haworth N. Powered two wheelers in a changing world-challenges and opportunities. Accid Anal Prevent. 2012;44(1):12–8. https://doi.org/10.1016/j.aap.2010.10.031.
2. World Health Organization: WHO Global Status Report on Road Safety 2018. World Health Organization, Geneva 2018; https://www.who.int/publications/i/item/9789241565684
3. Destatis: Verkehrsunfälle 2021 (in German). Verkehr **Fachserie 8 Reihe 7** 2022; https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/verkehrsunfaelle-jahr-2080700217004.pdf?__blob=publicationFile

4. Savino G, Lot R, Massaro M, Rizzi M, Symeonidis I, Will S, Brown J. Active safety systems for powered two-wheelers: a systematic review. Traffic Injury Prevent. 2020;21(1):78–86. https://doi.org/10.1080/15389588.2019.1700408.

5. Dean M, Haus SH, Sherony R, Gabler HC. Potential crash benefits of motorcycle-detecting automatic emergency braking systems. In: proceedings of the IRCOBI conference, Munich, Germany, 2021; 39– 50 . http://www.ircobi.org/wordpress/downloads/irc21/pdf-files/2115.pdf

6. Savino G, Pierini M, Fitzharris M. Motorcycle active safety systems: assessment of the function and applicability using a population-based crash data set. Traffic Injury Prevent. 2019;20(4):406–12. https://doi.org/10.1080/15389588.2019.1594795.

7. Maier S, Doléac L, Hertneck H, Stahlschmidt S, Fehr J. Evaluation of a novel passive safety concept for motorcycles with combined multi-body and finite element simulations. In: Proceedings of the IRCOBI Conference, Munich, Germany, 2020;250–265. https://doi.org/10.18419/opus-11899. IRCOBI

8. Autoliv: Annual and Sustainability Report 2021. Autoliv Inc., Stockholm, Sweden 2022; https://www.autoliv.com/sites/default/files/2022-04/Autoliv_Annual_Sustainability_Report_2021_1.pdf

9. Kobayashi Y, Makabe T. Crash detection method for motorcycle airbag system with sensors on the front fork. In: Proceedings of the 23rd international technical conference on the enhanced safety of vehicles (ESV), Seoul, Korea 2013; http://www-esv.nhtsa.dot.gov/Proceedings/23/isv7/main.htm

10. Kuroe T, Namiki N, Iijima S. Exploratory study of an airbag concept for a large touring motorcycle: Further research second report. In:19th International Technical Conference on the Enhanced Safety Vehicles 2005;. https://www-esv.nhtsa.dot.gov/Proceedings/19/05-0316-O.pdf

11. Engel A. Airbag Für Motorisierte Zweiräder: Bericht zum Projekt 1.8903 der Fahrzeugtechnischen Forschung des Bundesministers Für Verkehr (in German). Forschungsberichte des Bundesministers für Verkehr, Bereich Fahrzeugtechnik ; 8. Bundesanst. für Straßenwesen, Bergisch Gladbach 1992;

12. Daub S. Untersuchung von Sensorkonzepten zur Unfallerkennung von Motorrädern (in German). Bachelor Thesis BSC-137, Institute for engineering and computational mechanics, University of Stuttgart 2021;

13. Leschke A. Algorithm concept for crash detection in passenger cars. Springer, Wiesbaden, Germany 2020; https://doi.org/10.1007/978-3-658-29392-5

14. Theofilatos A, Chen C, Antoniou C. Comparing machine learning and deep learning methods for real-time crash prediction. Transp Res Record. 2019;2673(8):169–78. https://doi.org/10.1177/0361198119841571.

15. Huang T, Wang S, Sharma A. Highway crash detection and risk estimation using deep learning. Accid Anal. 2020;135:105392. https://doi.org/10.1016/j.aap.2019.105392.

16. Yang Y, He K, Wang Y.-p, Yuan Z.-z, Yin Y.-h, Guo M.-z. Identification of dynamic traffic crash risk for cross-area freeways based on statistical and machine learning methods. Phys A Statist Mech Appl. 2022;595:127083. https://doi.org/10.1016/j.physa.2022.127083.

17. Rezapour M, Molan AM, Ksaibati K. Analyzing injury severity of motorcycle at-fault crashes using machine learning techniques, decision tree and logistic regression models. Int J Trans Sci Technol. 2020;9(2):89–99. https://doi.org/10.1016/j.ijtst.2019.10.002.

18. Aloul F, Zualkernan I, Abu-Salma R, Al-Ali H, Al-Merri M. iBump: smartphone application to detect car accidents. Comput Electr Eng. 2015;43:66–75. https://doi.org/10.1016/j.compeleceng.2015.03.003.

19. Condro N, Li M-H, Chang R. Motosafe: active safe system for digital forensics of motorcycle rider with android. Int J Inf Electron Eng. 2012;2(4):612–6. https://doi.org/10.7763/IJIEE.2012.V2.171.

20. Mian SA. Development of a smartphone-based crash notification system for motorcycle drivers using machine learning 2021; https://odr.chalmers.se/server/api/core/bitstreams/bc73284d-ea09-4506-b8ed-b7f7d74d3ff9/content

21. Deng L. Development of a crash detection algorithm for motorcycle drivers using machine learning 2021; https://odr.chalmers.se/bitstreams/0e9f7971-105e-4afd-a49e-1335865196ab/download

22. Choi JG, Kong CW, Kim G, Lim S. Car crash detection using ensemble deep learning and multimodal data from dashboard cameras. Expert Syst Appl. 2021;183:115400. https://doi.org/10.1016/j.eswa.2021.115400.

23. Hiemer M. Model based detection and reconstruction of road traffic accidents. Universitätsverlag Karlsruhe, Karlsruhe 2005; . https://doi.org/10.5445/KSP/1000001370

24. Maier S, Fehr J. Efficient simulation strategy to design a safer motorcycle. Multibody System Dynamics, 2023; 1–28 https://doi.org/10.1007/s11044-023-09879-8

25. McKay M, Conover W. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics. 1979;21:239–45. https://doi.org/10.1080/00401706.2000.10485979.

26. Rodegast P, Maier S, Kneifl J, Fehr J. Simulation data from motorcycle sensors in operational and crash scenarios. https://doi.org/10.18419/darus-3301

27. Liu K, Yamamoto T, Morikawa T. Impact of road gradient on energy consumption of electric vehicles. Transp Res Part D Transp Environ. 2017;54:74–81. https://doi.org/10.1016/j.trd.2017.05.005.

28. Corno M, Savaresi SM, Tanelli M, Fabbri L. On optimal motorcycle braking. Control Eng Pract. 2008;16(6):644–57. https://doi.org/10.1016/j.conengprac.2007.08.001.

29. Hauser J, Saccon A. Motorcycle modeling for high-performance maneuvering. IEEE Control Syst Magaz. 2006;26(5):89–105. https://doi.org/10.1109/MCS.2006.1700047.

30. ISO13232-2:2005: ISO13232-2:2005, Motorcycles - Test and Analysis Procedures for Research Evaluation of Rider Crash Protective Devices Fitted to Motorcycles., International Organization for Standardization, Geneva, Switzerland 2005;154

31. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12:2825–30 (https://dl.acm.org/doi/pdf/10.5555/1953048.2078195).