Machine Learning Based Prediction of Crash Severity Distributions for Mitigation Strategies

Marcus Müller and Michael Botsch
Technische Hochschule Ingolstadt, Germany
Email: {marcus.mueller, michael.botsch}@thi.de

Dennis Böhmländer
AUDI AG, Ingolstadt, Germany
Email: dennis.boehmlaender@audi.de

Wolfgang Utschick
Technische Universität München, Germany
Email: utschick@tum.de

***Abstract*—In road traffic, critical situations pass by as quickly as they appear. Within the blink of an eye, one has to come to a decision, which can make the difference between a low severity, high severity or fatal crash. Because time is important, a machine learning driven Crash Severity Predictor (CSP) is presented which provides the estimated crash severity distribution of an imminent crash in less than 0.2ms. This is $63 \cdot 10^3$ times faster compared to predicting the same distribution through computationally expensive numerical simulations. With the proposed method, even very complex crash data, like the results of Finite Element Method (FEM) simulations, can be made available ahead of a collision. Knowledge, which can be used to prepare occupants and vehicle to an imminent crash, activate and adjust safety measures like airbags or belt tensioners before of a collision or let self-driving vehicles go for the maneuver with the lowest crash severity. Using a real-world crash test it is shown that significant safety potential is left unused if instead of the CSP-proposed driving maneuver, no or the wrong actions are taken.**

***Index Terms*—crash severity, vehicle safety, reliable prediction, machine learning**

I. Introduction

Crash severity in vehicle collisions mainly depends on the kinetic energy of the crash participants. To reduce the crash severity, reducing the forces acting on the occupants during a crash has been the goal of vehicle safety since its start in the early 1950s. One possibility to achieve this goal are structural measures like crumble zones, airbags or seat belts, which spread the forces experienced by the occupants over a longer time and thereby reduce the peak forces on them. These so-called Passive Safety measures had a huge impact on vehicle safety, pushing the number of fatalities on German roads from an all-time high of 21.332 cases in the year of 1970 down to 3.206 deaths in 2016, although the number of registered vehicles has tripled within this time [1].

Despite the success of Passive Safety, there are physical limits, such as in a high-speed crash of a small vehicle with a heavy truck, which no structural measure alone can overcome. At this point, Active Safety applications hook up, trying to avoid a collision by supporting proper braking or steering. Through exteroceptive sensors like lidar, radar, or camera, modern vehicles perceive their surroundings like other vehicles, pedestrians, or the road infrastructure. Advanced perception techniques enable developers of vehicle safety functions to recognize, rate and react to critical situations as early as a threat becomes visible to the sensors.

Figure 1. Crash test with dummy vehicle on the CARISSMA outdoor facility at Technische Hochschule Ingolstadt

This is often far ahead of the moment the driver gets aware, extending the available timeframe for countermeasures by a significant amount of time. One system utilizing this principle is the Autonomous Emergency Braking (AEB) [2]. The AEB automatically

Manuscript received October 16, 2017; revised January 24, 2018.

doi: 10.12720/jait.9.1.15-24

brakes the own vehicle (Ego) if the time gap to the leading vehicle (Object), the so-called Time-to-Collision (TTC), falls below a specified limit. According to a 2008 study by the European Commission, the AEB prevents an estimated 5,000 fatalities and 50,000 severe injuries in Europe every year [3], [4]. Limitations of the AEB are that only obstacles in a small sector in front of the Ego-vehicle are considered and braking is the only possible action to be performed.

In this paper, a machine learning based system is proposed, which predicts the crash severity distributions for several Ego-maneuvers, taking actions like steering, accelerating, braking or combinations of these into account. This allows other safety systems or algorithms, such as the trajectory planner of a self-driving car, to aim for the best driving maneuver. Knowing in advance that a sever crash, e. g. the side crash from Fig. 1 is going to happen also allows to lower the airbag activation thresholds, reducing the deployment time and, as a result, increasing the safety potential [5]. Based on the CSP-prediction, airbags might even be fired ahead of a collision if this helps to reduce the crash severity. Adjusting the belt tensioner to a previously known crash can minimize the forces an occupant is exposed to due to the pyrotechnical activation of the actuator [5]. Furthermore, an early activation is also a key requirement for future actuators like the exterior airbag, reversible electrical belt tensioner or very large indoor airbags, used in vehicles with innovative spacious interior designs. These applications all have in common that they take longer for activation than an average crash lasts and thus need to be fired beforehand.

In Section II, related work from the field of crash severity prediction is discussed. Section III describes the simulation framework used to generate the training data for the CSP. In Section IV, three examples of crash severity estimation are presented, one utilizing a rule-based approach build with the help of a FEM crash database, and another using a mass-spring-model. Section V deals with the machine learning background of the CSP for which the results are presented in Section VI. Section VII finally concludes the paper with a summary and a discussion about aspects of future work.

Throughout this paper, vectors and matrices are denoted by lower and upper case bold letters. A lower-case bold letter represents a column vector.

II. Related Work

The Frontal Crash Criterion (FCC) [6], the Head Injury Criterion (HIC) [7] and the Occupant Load Criterion (OLC) [7] are commonly used crash severity metrics. They allow comparing different collisions by providing figures with no unit, derived from simplified mechanical models based on the acceleration during a crash. The metrics were designed for frontal crash scenarios and thus might not be used for other crash types, like side or rear-end collisions.

When it comes to crash severity prediction, two main methodologies need to be distinguished. The first utilizes physical models to calculate the future car movements and numerically estimate the crash consequences, while in the second, statistical methods are used for statements about the probability of certain crash severities. Representatives of the first approach are [5] and [8], where in both works a combination of a vehicle dynamics and a collision-model [9] is used, to anticipate the severity of an imminent crash. On the other side, it also has been shown in [10], [11] and [12], that the crash severity of a collision can be classified using statistical learning methods. With [13] an approach, combining physical and statistical models for crash severity prediction has been presented.

III. Simulation Framework

Creating a statistical model requires comprehensive knowledge about the respective domain to be described. In the case of the CSP, knowledge about pre-crash situations with their corresponding crash constellations and severities is required. Crash databases like GIDAS [14] provide valuable real-world information about a large number of collisions, but still cannot satisfy the demand of a system, which needs to know how a crash changes when the driver reactions vary. Because such a system can easily require millions of crash observations, getting better with an even further growing number, simulations are currently the only feasible way to get access to sufficient data. In this section, the simulation framework used to generate the training data for the Crash Severity Predictor is explained.

A. Generation of Critical Traffic Scenarios

The so-called Accident Hypothesis Framework (AHF) consists of the open source traffic simulator SUMO [15] and a self-developed Matlab component. In the SUMO part, a traffic simulation of the city of Bologna [16] is carried out. Interfacing with Matlab via the TraCi4Matlab wrapper [17], a rule-based selection of potential crash candidates is performed. If a candidate is found, the SUMO simulation is paused, and the current situation of the candidate vehicles is shifted from SUMO to Matlab. In Matlab, the desired path of one vehicle is manipulated to match one of five randomized maneuver templates, four of them depicted in Fig. 2. This step of introducing a driving mistake is necessary, as in SUMO no accidents occur.

The randomized maneuvers are applied to the changing road and vehicle constellations taken from SUMO, resulting in unique traffic scenarios. With all randomness involved, the use of a Two-Track-Model (TTM) [13], [18]-[20] ensures that all simulated trajectories are physically plausible. The fact, that realistic road networks are used in the process, accounts for the circumstance, that the geometry of a road network carries *a priori* knowledge about the probabilities of certain crash constellations [21], [22].

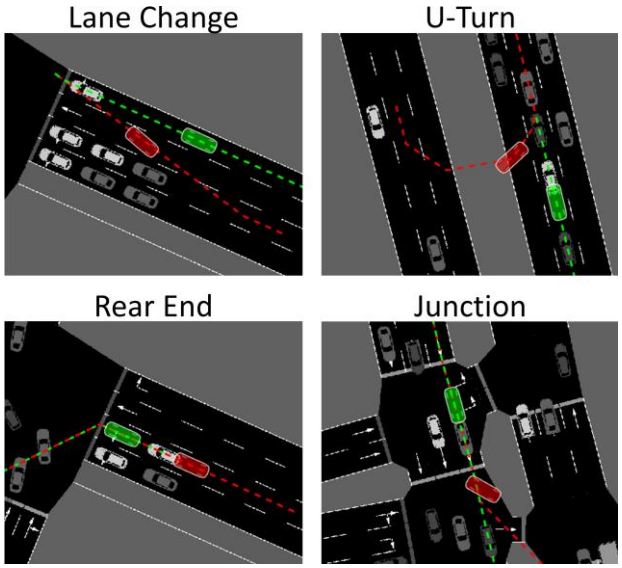


Figure 2. Accident hypothesis framework maneuver templates

B. Crash Severity Distribution

The goal of the AHF is the generation of crash severity distributions for a large number of pre-crash scenarios. As e.g. 500ms before a collision it is unclear, how the drivers are going to react, it cannot clearly be stated what crash and thus, what crash severity is going to appear. A probabilistic description of the situation is needed. Therefore, the AHF takes a pre-crash situation and anticipates with the TTM how the situation evolves, considering hundreds of maneuvers for both, the Ego- and the Object-vehicle. A maneuver is defined as the combination of a steering and an acceleration instruction like [-15, -0.6] for steering 15 ° right and decelerating with 60 % of the vehicle-specific maximum deceleration. A usual set of maneuvers contains the combinations of ten steering and ten acceleration instructions, ranging from extreme steering or accelerating to no-change. This gives 100 maneuvers per vehicle and 10.000 possible trajectory combinations for two vehicles, assuming that both use the same set of maneuvers. An Unavoidability Detector [13] checks whether all combinations end in a crash and rejects those scenarios with possible evasion trajectories. Situations, where a crash is avoidable, are rejected because an activation of irreversible safety systems is not desired in these cases. Typically, unavoidability occurs at TTCs of around 150 up to 800ms. When a crash is found unavoidable, all 10.000 trajectory combinations lead to one of 10.000 crash constellations. The pre-crash situation all the maneuvers start from is identified by $s \in \{1,..S\}$, where S denotes the total number of different pre-crash situations processed during a simulation run.

C. Data Structure

As stated above, a single unavoidable pre-crash situation s can lead to several different crash constellations, which, in turn, can have very different crash severities. For that reason, pairs of Ego- and Object-trajectories are evaluated until all crash severities

$$\mathbf{CS}^s = \begin{bmatrix} \mathrm{cs}_{11}^s & \cdots & \mathrm{cs}_{1M}^s \\ \vdots & \ddots & \vdots \\ \mathrm{cs}_{N1}^s & \cdots & \mathrm{cs}_{NM}^s \end{bmatrix} \in \mathbb{R}^{N\times M} \quad (1)$$

of the current pre-crash situation s are determined, with N and M denoting the number of considered Ego- and Object-maneuvers. Thus, the matrix element cs_{nm}^s is the crash severity, which appears for the trajectory pair resulting from the n^{th} Ego- and the m^{th} Object-maneuver, starting from the pre-crash situation s. For each new pre-crash situation adopted from SUMO, a feature vector $\mathbf{d}^s$ is generated and added as a new row entry to the training database $\boldsymbol{\mathcal{D}}$:

$$\mathbf{d}^s = [\mathbf{preCrash}^{s,\mathrm{T}}, \mathbf{anticipate}^{s,\mathrm{T}}, \mathbf{labels}^{s,\mathrm{T}}]^{\mathrm{T}} \quad \mathbf{d}^s \in \mathbb{R}^{L\times 1}, \quad (2)$$

The vector $\mathbf{d}^s$ is made up of the three elements,

$$\mathbf{preCrash}^s = \left[\boldsymbol{\xi}_{\mathrm{Ego}}^{\mathrm{T}}(\mathrm{t}_{\mathrm{pc}}^s), \boldsymbol{\xi}_{\mathrm{Obj}}^{\mathrm{T}}(\mathrm{t}_{\mathrm{pc}}^s)\right]^{\mathrm{T}} \in \mathbb{R}^{2F\times 1} \quad (3)$$

$$\mathbf{anticipate}^s = \left[\boldsymbol{\xi}_{\mathrm{Ego}}^{\mathrm{T}}(\mathrm{t}_0^s), \boldsymbol{\xi}_{\mathrm{Obj}}^{\mathrm{T}}(\mathrm{t}_0^s)\right]^{\mathrm{T}} \in \mathbb{R}^{2F\times 1} \quad (4)$$

$$\mathbf{labels}^s = \mathbf{distribCS}^s \text{ OR } \boldsymbol{p}_{\mathrm{fire}}^s \quad (5)$$

with $\boldsymbol{\xi}_{\mathrm{Ego}}^{\mathrm{T}}(\mathrm{t}_{\mathrm{pc}}^s)$ and $\boldsymbol{\xi}_{\mathrm{Obj}}^{\mathrm{T}}(\mathrm{t}_{\mathrm{pc}}^s)$ being the states vectors of the Ego- or the Object-vehicle each of length F at a point in time $\mathrm{t}_{\mathrm{pc}}^s$ of the s^{th} scenario, after the unavoidability but before the crash time instance t_0. The first two elements $\mathbf{preCrash}^s$ and $\mathbf{anticipate}^s$ are state vectors which describe the present and one of the many possible future states of both vehicles.

The state vector

$$\boldsymbol{\xi}_{\mathrm{Ego}}(\mathrm{t}) = [\mathrm{x}_{\mathrm{Obj}}(\mathrm{t}), \mathrm{y}_{\mathrm{Obj}}(\mathrm{t}), \psi_{\mathrm{Obj}}(\mathrm{t}), \mathrm{v}_{\mathrm{Ego}}(\mathrm{t}) \quad \mathrm{v}_{\mathrm{Obj}}(\mathrm{t}), \mathrm{a}_{x,\mathrm{Ego}}(\mathrm{t}), \mathrm{a}_{y,\mathrm{Ego}}(\mathrm{t}), \ldots]^{\mathrm{T}} \in \mathbb{R}^{F\times 1} \quad (6)$$

describes the situation of the Ego-vehicle with information such as the position of the Object-vehicle $[\mathrm{x}_{\mathrm{Obj}}^s, \mathrm{y}_{\mathrm{Obj}}^s]$ in the Ego body frame, the angle between the heading of both vehicles ψ, the velocity v or the acceleration a. Accordingly, $\boldsymbol{\xi}_{\mathrm{Obj}}(\mathrm{t})$ describes the situation of the Object-vehicle from an Object-vehicle perspective, with the quantities of $\boldsymbol{\xi}_{\mathrm{Obj}}$ being described in the Object body frame. While $\mathbf{preCrash}^s$ represents the state vector at t_{pc} e.g. 500ms before a crash, $\mathbf{anticipate}^s$ contains the equivalent description for the start of the crash that would occur when both cars maintain the direction and velocity they have at t_{pc} (no-change assumption). The length F of $\boldsymbol{\xi}_{\mathrm{Ego}}$ or $\boldsymbol{\xi}_{\mathrm{Obj}}$ represents the number of features that are used to describe the state of either Ego or Object. In total, there are more than 150 features available in the AHF from which a small subset of up to 29 features is selected. The ten most important features are presented in Section VI.E.

The vector $\mathbf{labels}^s$ contains the values, which later shall be predicted by the CSP. Depending on what description of the crash severity is desired (see Section IV), $\mathbf{labels}^s$ is either $\mathbf{distribCS}^s$ or $\boldsymbol{p}_{\mathrm{fire}}^s$. While $\boldsymbol{p}_{\mathrm{fire}}^s$ represents the probability for a crash severity that requires an airbag activation, as explained in Section IV.B, $\mathbf{distribCS}^s$ stands for quantities that describe the estimated

distribution of an arbitrary crash severity measure to be predicted. As it would be inefficient to learn all elements in $\mathbf{CS}^s$ to obtain its distribution, the main distribution characteristics like the minimum or maximum crash severity are learned instead:

$$\mathbf{distribCS}^s = [\mathbf{cs}_{\min}^{s,\mathrm{T}}, \mathbf{cs}_{\mathrm{p25}}^{s,\mathrm{T}}, \mathbf{cs}_{\mathrm{med}}^{s,\mathrm{T}}, \ldots \mathbf{cs}_{\mathrm{p75}}^{s,\mathrm{T}}, \mathbf{cs}_{\max}^{s,\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{5N\times 1} \quad (7)$$

$$\mathbf{cs}_{\min}^s = [\min(\mathrm{cs}_{11}^s..\mathrm{cs}_{1M}^s), \min(\mathrm{cs}_{21}^s..\mathrm{cs}_{2M}^s), \ldots \min(\mathrm{cs}_{N1}^s..\mathrm{cs}_{NM}^s)]^{\mathrm{T}} \in \mathbb{R}^{N\times 1} \quad (8)$$

$$\mathbf{cs}_{\mathrm{p25}}^s = [\mathrm{p25}(\mathrm{cs}_{11}^s..\mathrm{cs}_{1M}^s), \mathrm{p25}(\mathrm{cs}_{21}^s..\mathrm{cs}_{2M}^s), \ldots \mathrm{p25}(\mathrm{cs}_{N1}^s..\mathrm{cs}_{NM}^s)]^{\mathrm{T}} \in \mathbb{R}^{N\times 1} \quad (9)$$

where $\min(\mathrm{cs}_{11}^s..\mathrm{cs}_{1M}^s)$ denotes the minimum value of the first row of $\mathbf{CS}^s$ and p25 denotes the 25th percentile of the corresponding row. The median $\mathbf{cs}_{\mathrm{med}}^s$, 75th percentile $\mathbf{cs}_{\mathrm{p75}}^s$, and maximum $\mathbf{cs}_{\max}^s$ are calculated equivalently.

As from an Ego-safety-system-perspective, the Ego-maneuver is the only means to influence the outcome of a crash all elements in $\mathbf{distribCS}^s$ are calculated on a per-ego-maneuver basis. That is why $\mathbf{cs}_{\min}^s$, $\mathbf{cs}_{\mathrm{p25}}^s$ etc. are $N \times 1$ vectors, containing one value for each Ego-maneuver. With $\mathbf{distribCS}^s$ it is possible to say, which Ego-maneuver leads to which minimum, median and maximum crash severity together with the 25th and 75th percentile of the estimated distribution. This allows a prediction of the expected crash severity for the Ego-vehicle, despite the fact that the Object-maneuver is unknown. A statement about how reliable the prediction is can be made up based on the interquartile range

$$\boldsymbol{iqr}^s = \mathbf{cs}_{\mathrm{p75}}^s - \mathbf{cs}_{\mathrm{p25}}^s, \in \mathbb{R}^{N\times 1} \quad (10)$$

The smaller the interquartile range of the predicted distribution for a particular Ego-maneuver, the smaller the variations in the crash severity and the higher the chance that a crash severity close to the predicted median occurs.

IV. Crash Severity Estimation

To fill the previously introduced scenario description $\mathbf{d}^s$ with the crash severity measures of $\mathbf{distribCS}^s$, the crash severity has to be estimated first. The used crash severity measure is interchangeable. A selection of three exemplary measures is presented in this section.

A. Relative Velocity

It is known that the relative velocity between two vehicles

$$\mathrm{v}_{\mathrm{rel}} = \sqrt{\left(\mathrm{v}_{\mathrm{Ego},x} - \mathrm{v}_{\mathrm{Obj},x}\right)^2 + \left(\mathrm{v}_{\mathrm{Ego},y} - \mathrm{v}_{\mathrm{Obj},y}\right)^2} \quad (11)$$

correlates with the injury risk of the occupants [9]. Thus, predicting the expected crash severity for a certain maneuver can be achieved, by anticipating the future car movements with the TTM and measure the relative velocity $\mathrm{v}_{\mathrm{rel,t0},nm}^s$ at t_0, the moment the crash, caused by the n^{th} and m^{th} Ego- and Object-maneuver, begins. One drawback of $\mathrm{v}_{\mathrm{rel,t0},nm}^s$ is that it neither reflects the influence of the vehicle orientations nor the influence of the collision point. The difference in the crash severity between e.g. a front and a side crash, which arises from the lack of crumble zones in a side collision, is neglected if both constellations have the same $\mathrm{v}_{\mathrm{rel,t0}}$.

B. Airbag Activation Probability

For the next crash severity measure the matrix

$$\mathbf{Fire}^s \in \mathbb{R}^{N\times M}, \mathrm{fire}_{nm}^s \in \{0,1\} \quad (12)$$

is required, with its elements fire_{nm}^s indicating whether an airbag was fired or not in the corresponding crash constellation identified by the Ego- and Object-maneuvers n and m. Using $\mathbf{Fire}^s$, the conditional probability for the event of an airbag activation $\lambda = 1$, given a certain Ego-maneuver n executed starting from the pre-crash situation $\mathbf{preCrash}^s$ can be calculated:

$$p_{\mathrm{fire},n}^s = P(\lambda = 1|\mathbf{preCrash}^s) \approx \frac{1}{M}\sum_{m=1}^{M} \mathrm{fire}_{nm}^s, n \in \{1,..N\} \quad (13)$$

and for all Ego-maneuvers:

$$\boldsymbol{p}_{\mathrm{fire}}^s = \left[p_{\mathrm{fire},1}^s, \ldots, p_{\mathrm{fire,N}}^s\right]^{\mathrm{T}} \quad (14)$$

A higher probability $p_{\mathrm{fire},n}^s$ means that for the n^{th} Ego-maneuver it is more likely to face a situation where an airbag is required, indicating a higher crash severity.

Whether a crash is a fire or no-fire case is thereby determined with the help of the so-called Labeler.

The Labeler decides whether a crash constellation makes the use of one or more airbags necessary or not. It does so because an automated way to differentiate fire from no-fire cases is required when millions of crash constellations generated by the Accident Hypothesis Framework shall be processed. With this, it replaces a human expert, which would analyze each collision and give it either the label fire or no-fire, depending on the expectation of the expert on whether a given situation will make the use of one or more airbags necessary.

To automate this step, a FEM-database with 1,487 highly detailed simulations of car-to-car collisions, also containing the information about which airbags were fired, is used. By analyzing the database, a ruleset could be defined to correctly classify 99.4 % of the database collisions. The remaining nine entries were found to be outliers, caused by numerical issues during the FEM-simulations. Fig. 3 illustrates the different steps to finally separate fire from no-fire cases in an easily interpretable, two-dimensional space. First, the crash constellations are divided into three clusters, depending on whether they describe a Front-Front, Rear-Front or Front-Rear collision, whereas Front is defined as the frontal 50 % of the vehicle length and Rear as the remaining rear 50 % of the vehicle length. A Front-Rear collision stands for the Ego-vehicle hitting with its front the rear of the Object-vehicle. Rear-Rear collisions are very unlikely as usually, at least one car drives forward and thus are not present in the database.

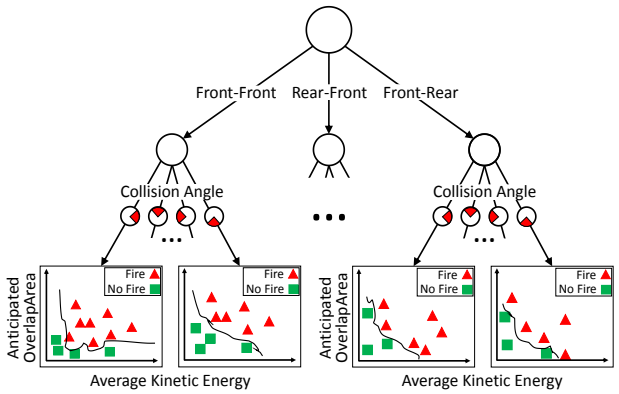


Figure 3. Interpretable decision tree like fire/no-fire labeler

In a second step, the crash constellations are separated by the angle between the vehicle headings. The four cases: [-45 °, 45 °], [45 °, 135 °], ([135 °, 180 °] or [-135 °, 180 °]) and [-135 °, -45 °] are distinguished. The previous two steps result in 12 clusters, which finally can be visualized in the space spanned by the Average Kinetic Energy

$$\text{ake} = \frac{1}{2}(e_{\text{kin,Ego}} + e_{\text{kin,Obj}}) \tag{15}$$

and the Anticipated Overlap Area

$$\text{aoa} = \mathcal{O}\left(\boldsymbol{\xi}_{\text{Ego}}(t_0 + 50\,\text{ms}), \boldsymbol{\xi}_{\text{Obj}}(t_0 + 50\,\text{ms})\right) \tag{16}$$

where $\mathcal{O}$ returns the extrapolated overlap in [m²] of both vehicles 50ms after t_0, assuming linear vehicle movement without interaction between the cars. The aoa is an important feature because it combines the vehicle shapes, velocities, orientations, and relative positions. The ake on the other side also uses the velocities and adds the information about the vehicle masses, as $e_{\text{kin}} = 0.5 \cdot mv^2$.

Finally, a decision boundary to separate fire from no-fire cases is manually applied to each cluster.

C. Mass-Spring-Model

Both previously presented metrics do not model the in-crash phase but instead directly map from a crash constellation to either $v_{\text{rel,t0}}$ or λ. The mass-spring-model approach [13] in contrast considers the in-crash phase by simulating the interactions of two crash participants. The cars are represented through their masses m_1 and m_2 as well as their positions r_1 and r_2 along the axis of their relative movement.

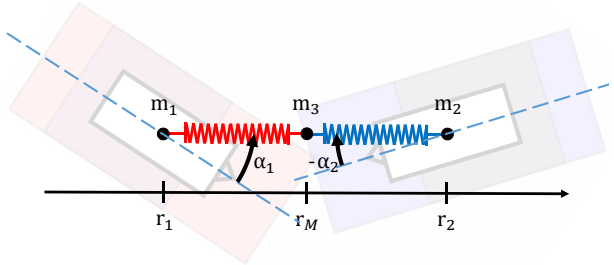


Figure 4. Crashing vehicles with superimposed mass-spring-model

The car structures are modeled through two springs with their primary characteristic being their stiffnesses k_1 and k_2. A third virtual mass m_3 with position r_M completes the mass-spring model by connecting the two springs and thereby, forming a line of three masses interlinked by the two springs, as shown in Fig. 4.

The vehicle velocities $v_1(t_0)$ and $v_2(t_0)$ are known and thus, the compression of the springs and thereby the resulting forces

$$\begin{aligned} f_1(t) &= k_1\big(r_M(t) - r_1(t)\big) \\ f_2(t) &= k_2\big(r_2(t) - r_M(t)\big) \end{aligned} \tag{17}$$

can be calculated. Given the forces also the accelerations in longitudinal and lateral vehicle direction

$$\mathbf{a}_p(t) = \frac{f_p(t)}{m_p}\begin{bmatrix}\cos\left(\alpha_p(t)\right) \\ \sin\left(\alpha_p(t)\right)\end{bmatrix}, p \in \{1,2\} \tag{18}$$

are known. With the accelerations, the velocities, in turn, can be updated and so on. A detailed description of the model was presented in [13]. The anticipated crash pulse $\mathbf{a}_p$ carries valuable information about the crash severity and e. g. can be used to calculate the OLC or other pulse-induced metrics. Because the OLC works only for frontal collisions, a prototypical crash severity measure which draws on the results from the mass-spring-model, accepting any possible crash constellation is suggested below.

D. Prototypical Crash Severity Measure

The following prototypical crash severity measure is proposed

$$\text{pcs} = f_{\text{o2c}} = \frac{1}{2l_{\text{oi}}} m_{\text{occ}} v_{\text{o2c}}^2(t_{\text{oi}}) \tag{19}$$

$$v_{\text{o2c}}(t) = \int_{t_0}^{t} \|\mathbf{a}_p(t)\| dt \tag{20}$$

with $v_{\text{o2c}}(t_{\text{oi}})$ being the relative occupant-to-car (o2c) velocity with which the occupant hits the vehicle interior at t_{oi}, l_{oi} being the deceleration streak over which the occupant is decelerated in interaction with the interior and m_{occ} being the occupant mass. Equation (20) holds given that the occupant is not decelerated by any restraint system like seatbelt or airbag. Furthermore, it is assumed that the whole kinetic energy of the occupant

$$e_{\text{kin,o2c}} = \frac{1}{2} m_{\text{occ}} v_{\text{o2c}}^2 \tag{21}$$

is transformed during the crash into the mechanical work

$$w_{\text{o2c}} = f_{\text{o2c}} l_{\text{oi}} \tag{22}$$

Thus, the pcs corresponds to the force f_{o2c}, experienced by an occupant when he hits the vehicle interior. Varying the deceleration streak l_{oi}, harder or softer parts of the interior can be modeled. Fig. 5 shows exemplarily how a vehicle in the AHF can be divided into different stiffness zones.

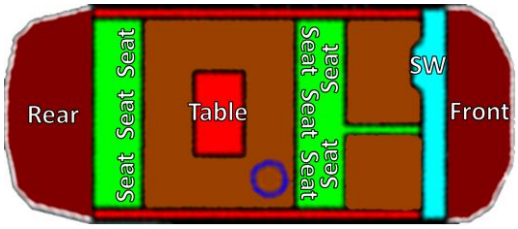


Figure 5. Soft (g), hard (b) and very hard (r) vehicle interior zones

Green zones represent soft interior (e.g. seats) whereas blue and red zones represent hard and very hard parts, like the steering wheel, the chassis or as in this example, a table how it might appear in the center of a modern, self-driving vehicle. The circle marks the moving occupant position.

To determine where and when an occupant hits the interior its relative displacement

$$\mathbf{d}_{o2c}(t) = \iint_{t_0}^{t} \mathbf{a}_p(t)\, dt \qquad (23)$$

needs to be calculated. While the orientation and the velocity of both vehicles might continuously change during the crash due to the crash forces, the occupants are assumed to be decoupled from the vehicles and thus, maintain the velocity and direction they have at t_0. This linear motion is continued until a collision of the occupant with one of the zones is detected and, as a consequence, $v_{o2c}(t_{oi})$ becomes known.

V. Reliable Crash Severity Prediction

The goal of the CSP is to predict the crash severity only milliseconds ahead of a collision. Evaluating thousands of trajectory pairs to obtain $\boldsymbol{p}^s_{\text{fire}}$ or $\mathbf{distribCS}^s$ as discussed in Section IV, is hardly feasible under the given time constraints. While some crash severity measures like $\boldsymbol{p}^s_{\text{fire}}$ or $v_{\text{rel,t0}}$ might be suitable for online evaluation, the estimation of the possible crash constellations for the given pre-crash situation remains as a computationally expensive step. It always has to be performed before a crash severity estimation can be carried out. This means that for a large number of trajectory pairs differential equations would need to be solved numerically in real-time. For that reason, machine learning is used to directly predict the crash severity instead. Methods like Random Forest [23], Multi Layer Perceptron [24], [25] or Support Vector Machine [26] have been tested against this problem, and in accordance with the results of other researchers dealing with similar problems [12], Random Forest was found to perform well regarding training speed and prediction accuracy.

The general idea is to use the AHF, as presented in Section III, to generate the data for a large number of situations during an offline (=outside the car) simulation session, lasting days, weeks or even months. This data is then used to train a so-called Random Jungle, composed of multiple independent Random Forest models, one for each desired target variable, like the minimum, median or maximum crash severity. The Random Jungle can then provide all these information online (=inside the car) within very short time. Especially compared to simulating all trajectory pairs online this is much faster as discussed in Section VI.F. For each pre-crash situation, the whole crash severity distribution is available during the AHF simulation and the characteristic variables like maxima, minima, etc. are stored and shall now be learned. The goal of this process is shown in Fig. 6: The prediction of a boxplot for $N = 5$ different Ego-maneuvers. For each of the five maneuvers a box, representing the crash severity distribution for the corresponding Ego-maneuver is depicted.

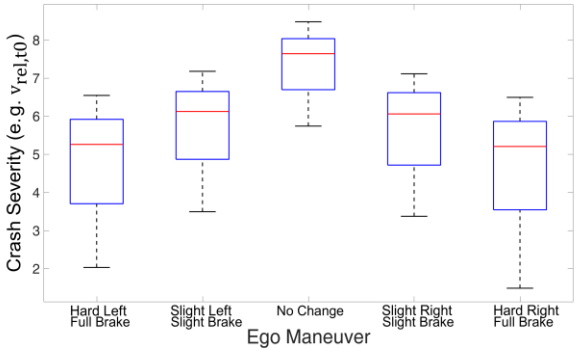


Figure 6. CSP result: 5 ego-maneuvers with their CS-distributions

The data required to plot all boxes was introduced as $\mathbf{distribCS}^s$ in Section III.C. To produce a result like the depicted one, 25 Random Forest models have to be trained, as each of the five maneuvers requires a cs_{min}, cs_{p25}, cs_{med}, cs_{p75}, and cs_{max}.

One benefit of having data like this available before a collision is to be able to adjust further steps to the given circumstances. It becomes clear what range of crash severity must be expected and which maneuver should be performed to mitigate the crash consequences as good as possible. The deviation of particular maneuvers can be seen, and probabilities for different events, like exceeding a certain severity can be derived.

VI. Results

A. AHF-Data

In the course of one week, a database $\boldsymbol{\mathcal{D}}$ with 275,594 different entries $\mathbf{d}^s$, each describing one pre-crash situation, has been generated. To speed up the process, a crash severity distribution with only $15 \times 15 = 225$ maneuvers was chosen, leading altogether to more than 62million simulated accidents. The 15 maneuvers evaluated for each car are [A]ccelerate, [B]rake and [C]ontinue, combined with the five steering actions 1: hard left, 2: left, 3: straight, 4: right and 5: hard right. The corresponding Ego-maneuvers are denoted as A1, A2, …, C5. Continue stands for maintaining the initial velocity. For validation purpose, the data was split into a distinct training set $\boldsymbol{\mathcal{L}}$ with cardinality $S_{\mathcal{L}} = 201{,}044$ and a test set $\boldsymbol{\mathcal{T}}$ with $S_{\mathcal{T}} = 74{,}550$. For the validation of the prototypical crash severity measure pcs two dedicated test and training sets with $S_{\mathcal{L},\text{pcs}} = 32{,}056$ and $S_{\mathcal{L},\text{pcs}} = 1{,}241$ have been created. This was done because

the pcs requires a more time-consuming AHF simulation, including the mass-spring-model for in-crash modeling.

B. Prediction Accuracy

The CSP Random Jungle consists of 15 individual Random Forest models to predict p_{fire} plus another 75 models for predicting either $v_{rel,t0}$ or the pcs. To evaluate the prediction accuracy of the CSP, the Mean Absolute Error

$$mae = \frac{1}{S}\sum_{s=1}^{S} |Y_s - T_s|, \text{with } S = |\mathcal{T}| \tag{24}$$

and the Pearson correlation coefficient

$$\rho = \frac{\text{cov}(Y,T)}{\sigma_Y \sigma_T} \tag{25}$$

are calculated for each model individually. Y_s denotes the prediction result produced by a model for the situation s, T_s denotes the correct target value or label of the corresponding test data and σ_Y, σ_T are the standard deviations of Y and T. Table I shows the results of both metrics for all 165 models. The first line of each box represents the mae, the second line the correlation coefficient ρ. It should be noted that the units of the mae vary with the prediction target which can be either the probability of p_{fire}, the velocity in [m/s] for $v_{rel,t0}$ or the force on an occupant in Newton for the pcs. It is shown in Table I that all three crash severity measures can successfully be learned by the CSP.

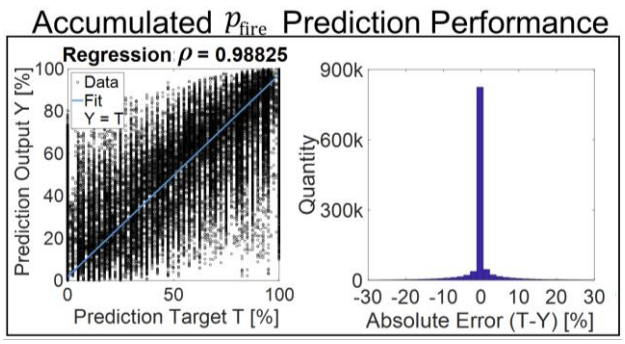
TABLE I. RANDOM JUNGLE PREDICTION ACCURACY

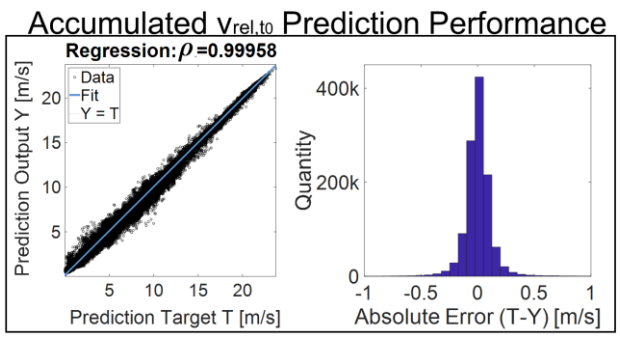
	p_{fire} [p]	$v_{rel,t0}$ [m/s]					pcs [N]				
		min	25th	med	75th	max	min	25th	med	75th	max
A1	0.027	0.156	0.150	0.148	0.153	0.155	289.1	307.7	341.7	397.3	477.3
	0.985	0.999	0.999	0.999	0.999	0.999	0.965	0.961	0.947	0.942	0.929
A2	0.026	0.155	0.150	0.152	0.154	0.156	286.7	313.9	357.8	427.0	473.5
	0.985	0.999	0.999	0.999	0.999	0.999	0.963	0.959	0.942	0.937	0.929
A3	0.024	0.151	0.150	0.151	0.154	0.157	278.9	325.5	385.7	410.6	497.0
	0.987	0.999	0.999	0.999	0.999	0.999	**0.980**	0.979	**0.973**	0.962	0.931
A4	0.025	0.149	0.148	0.148	0.151	0.156	304.2	293.3	381.3	422.7	510.6
	0.987	0.999	0.999	0.999	0.999	0.999	0.928	0.941	0.898	0.884	0.866
A5	0.025	0.150	0.148	0.148	0.154	0.154	291.7	300.2	374.9	438.2	516.8
	0.987	0.999	0.999	0.999	0.999	0.999	0.935	0.942	0.904	0.871	0.861
B1	0.023	0.153	0.118	0.096	0.106	0.120	**245.9**	279.5	352.5	452.8	550.8
	0.991	0.998	0.999	0.999	0.999	0.999	0.962	0.956	0.933	0.878	0.862
B2	0.023	0.148	0.117	0.096	0.105	0.122	266.9	277.1	362.2	457.1	521.7
	0.991	0.998	0.999	0.999	0.999	0.999	0.961	0.961	0.933	0.873	0.874
B3	**0.019**	0.143	**0.115**	**0.095**	0.105	0.122	297.3	**272.0**	**303.9**	**323.5**	**341.1**
	0.994	0.998	**0.999**	**0.999**	0.999	0.999	0.967	0.979	0.972	**0.967**	**0.962**
B4	0.022	0.150	0.120	0.097	**0.104**	**0.119**	281.5	313.3	371.5	394.5	435.0
	0.992	0.998	0.999	0.999	**0.999**	**0.999**	0.939	0.941	0.917	0.901	0.911
B5	0.022	0.149	0.119	0.095	0.104	0.120	267.0	294.1	353.1	389.5	442.5
	0.992	0.998	0.999	0.999	0.999	0.999	0.942	0.945	0.924	0.904	0.910
C1	0.027	0.130	0.122	0.125	0.132	0.140	299.5	307.4	364.0	449.4	500.4
	0.986	0.999	0.999	0.999	0.999	0.999	0.955	0.955	0.945	0.920	0.909
C2	0.027	0.130	0.122	0.126	0.133	0.138	286.4	309.5	396.2	460.6	506.9
	0.986	0.999	0.999	0.999	0.999	0.999	0.958	0.951	0.940	0.921	0.914
C3	0.022	**0.126**	0.120	0.125	0.129	0.140	321.3	278.1	343.6	397.1	484.7
	0.989	**0.999**	0.999	0.999	0.999	0.999	0.975	**0.980**	0.973	0.966	0.941
C4	0.026	0.129	0.121	0.122	0.131	0.137	305.0	319.5	347.1	398.2	514.0
	0.987	0.999	0.999	0.999	0.999	0.999	0.921	0.912	0.901	0.888	0.858
C5	0.026	0.130	0.119	0.124	0.129	0.141	308.7	306.0	346.3	420.2	506.2
	0.987	0.999	0.999	0.999	0.999	0.999	0.917	0.922	0.901	0.880	0.854
ø	**0.024**	**0.143**	**0.129**	**0.123**	**0.130**	**0.138**	**270.6**	**299.8**	**358.8**	**415.9**	**485.2**
	0.988	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.892**	**0.893**	**0.875**	**0.856**	**0.901**

The average mae for p_{fire} is 0.024 or 2.4%, The average correlation between true and predicted values of p_{fire} is 0.988. The values for $v_{rel,t0}$ in $\mathcal{L}$ range from 0 to 25.45m/s, with a mean of 6.55m/s. Thus, the average mae of 0.13 m/s is equivalent to an error of 2%. The average correlation is 0.99. For the prototypical crash severity pcs the values in $\mathcal{L}_{\text{pcs}}$ range from 0 to forces beyond 40,000N.

These values should be understood as relative comparison of different accidents rather than absolute forces, because the parameters for the mass-spring-model, like the deceleration streak l_{oi}, are chosen on a best guess basis to produce plausible results but are not validated yet. The average minimum pcs in $\mathcal{L}_{\text{pcs}}$ is 1830N and the average maximum pcs is 3308N. In relation to these values, the average mae(min) of 280N and mae(max) of 500N are equivalent to an error of 15.1%.

Fig. 7 shows visualizations of the accumulated prediction performances of all three predictor types. For p_{fire}, the plot shows the accumulated results of the 15 models for the 15 different Ego-maneuvers A1, …, C5. The other two plots contain the results of 75 models each, containing the predictions of min, p25, med, p75 and max for the 15 Ego-maneuvers. While the regression plot of $v_{rel,t0}$ appears clean, it is noticeable that the plots of pcs and p_{fire} have a more heterogeneous appearance. In the case of pcs, this comes from the result of the different models being accumulated, whereas in the case of p_{fire} it seems that a small percentage of the $15 \times S_{\mathcal{T}} = 1{,}118{,}250$ test instances with incorrect values is causing this visual result. On the right side of the figure, it can be seen, that the number of instances with an absolute error larger than 20% is almost invisible on the given scale. In absolute figures, it can be stated, that only 34,714 instances or 3.1% of $\mathcal{T}$ result in a mae larger than 20%.





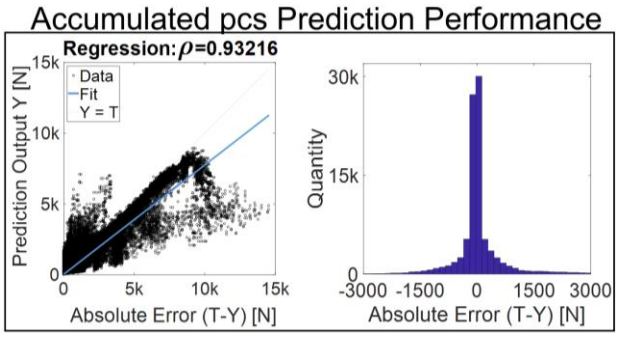


Figure 7. Regression plots and absolute error for p_{fire}, $v_{\text{rel,t0}}$ and pcs

C. System Behaviour on Real-World Data

Fig. 8 shows the results of the CSP for the real-world scenario from Fig. 1 with the test vehicle (red) crashing into the right side of a dummy vehicle (green). The Random Jungle is trained with data generated using the AHF. The trained models are then tested against data from the real world scenario. The different parameters like maximum or minimum crash severity are obtained by the CSP composed of 90 Random Forest models. The positions and yaw-angles of both cars are recorded using a Local Positioning Measurement System (LPM) with two transponders per vehicle. The data is synchronized and low-pass filtered before the velocity is obtained by deriving the position data. The vehicle shapes, masses, and stiffnesses (see Section IV.C) are taken from the set of available AHF-models, which match the real vehicles best.

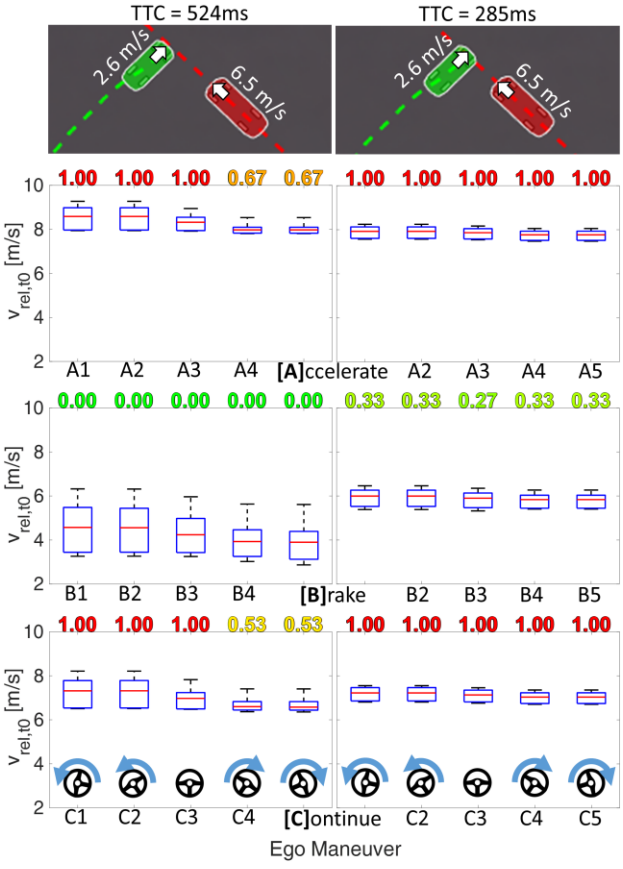


Figure 8. CSP with 90 RF-models applied on real world data

To illustrate how the chosen pre-crash situation evolves, the two time instances at a TTC of 524ms (left) and a TTC of 285ms (right) are presented in Fig. 8. The three depicted rows [A]ccelerate, [B]rake and [C]ontinue show the prediction of the crash severity measure $v_{\text{rel,t0}}$ for the possible Ego-maneuvers A1, …, C5. The colored numbers ranging from 0.00 to 1.00 represent the predicted results for p_{fire}.

It can be seen that between the worst Ego-maneuver A2 and the best maneuver B5 a difference of 6.5m/s or 23.4km/h is possible. Compared to the no-change case C3, still, a reduction in the range of 1-4m/s is possible. It is also shown that no airbag will be fired if the Ego-vehicle brakes, while in all the other cases the probability for an airbag activation is almost always 100%. Only for the maneuvers 4 and 5 (steer right) a p_{fire} of less than 100 % is possible. Steering right in general leads to a smaller crash severity according to the CSP. This makes sense, as steering right further decreases the relative velocity of both cars, now driving in a more similar direction. Thus, the predicted $v_{\text{rel,t0}}$ as well as the predicted p_{fire} seem plausible.

If no action is taken at TTC=524ms, it is apparent from Fig. 8 that the number of opportunities decreases over time. Good options available 524ms ahead of a crash might be lost 250ms later. This expectation is also confirmed by the results of the real-world experiment where both crash severity measures, $v_{\text{rel,t0}}$, and P_{fire}, are worse at a TTC of 285ms compared to a TTC of 524ms.

D. Delays

Delays arise at multiple points in the processing chain, like the sensors, data transmission or processing, and the actuators. It can be seen from Fig. 8 that delays have a huge impact on the effectivity of the CSP. A delay of 250ms is equivalent to the time difference between the left and the right side in the figure. This means that at a TTC of 524ms (left) the lowest crash severity the CSP could achieve by [B]raking is a collision with 3m/s, whereas 250ms later (right) only 5.8m/s can be achieved and thus a 10km/h or 43% faster accident occurs, based on the initial velocity of 6.5m/s. This means, that when the whole system, consisting of the vehicle, sensors, algorithms, and actuators from the moment it faces the left situation takes 250ms to react, the possible safety gain reduces to approximately 0.5m/s (right) rather than 3m/s (left) compared to the [C]ontinue case. Thus, it can be stated, that reducing the delays also has a beneficial impact on the effectivity of predicting systems in general and the CSP in particular.

E. Feature Importance

Each pre-crash situation s generated with the AHF is described by a feature vector $\mathbf{d}^s$ composed of over 150 elements. The whole feature vector was saved during simulation and analyzed afterwards regarding the importance of individual features. In order to determine the importance of a feature $d_i \in \mathbf{d}$, with i=1, 2, … L, the values of the feature under test are randomly permuted, and the impact on the prediction accuracy is recorded. Fig.

9 shows the ten most important features by crash severity type.

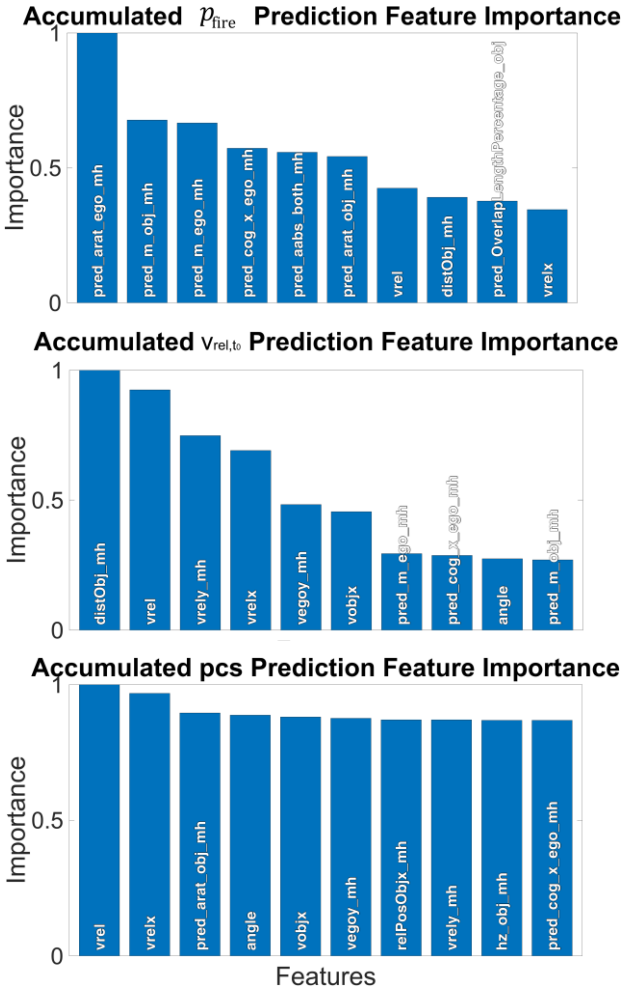


Figure 9. Most important features

Results are normalized by the most important feature. Features starting with "pred_" and/or ending on "_mh" describe the anticipated crash as explained for anticipate[s] in Section III.C. The abbreviations v, m, a, cog, hz, and dist stand for velocity, mass, area, center of gravity offset, hitzone, and distance. An anticipated mass pred_m results from the 50ms anticipation overlap percentage multiplied by the mass of the affected vehicle. The x- and y-axis point to the front and left of the Ego-vehicle. Arat means the ratio of overlap percentages and aabs is the absolute overlap in [m²].

As a result, it can be summarized, that $v_{rel,t0}$ and pcs mainly depend on velocity based features, whereas the prediction of p_{fire} relies on geometrical properties and more than the others on the vehicle masses. This makes sense as the CSP has learned the decision pattern of the Labeler and thus the three most important features perfectly represent the two-dimensional space introduced with the Equations (15) and (16).

F. Times

Training one Random Forest Model on $S_{\mathcal{L}} = 201{,}044$ instances takes approximately 64.34 seconds on an Intel i7 2.3GHz computer with 16 GB RAM. Thus, the Random Jungle with its 90 models takes around 96 minutes to train. As the models are independent, the work can be split to several machines to reduce the training time. Testing the Random Jungle against $S_{\mathcal{T}} = 74{,}550$ test instances takes 11.17 seconds or 150 µs per instance. Random Forest prediction can be parallelized with moderate hardware requirements. This theoretically allows predicting all 90 results of the CSP in parallel within 150 µs. Even if the prediction has to be performed on a single-core CPU and thus, the 90 predictions must be processed sequentially, a whole run takes only 13.5ms. This is still less than the 20ms cycle time of most sensors. So even without parallelization, the CSP can run in real-time in a usual vehicle environment. Simulating 225 trajectory pairs with the AHF, in contrast, takes 9.45 seconds and thus $63 \cdot 10^3$ times longer. While the time the Random Jungle takes for prediction remains constant independent on how many trajectory pairs are used for training and how complex their simulation is (e.g. FEM simulations), the time for the simulations linearly increases with the number of trajectory pairs and depending on the complexity of the used models (mass-spring-model vs. FEM).

VII. Conclusions and Future Work

Ahead of an unavoidable collision, taking the right actions decides about whether the crash severity will be high or low. A driver assistance system, which aims to react in order to mitigate the crash consequences, has to be fast and reliable at the same time. Two goals, which are hard to bring together, as usually a tradeoff between speed and precision has to be made.

This paper presents a machine learning driven approach, which makes the results of even highly complex crash severity simulations available within less than one millisecond. A system is presented, that allows to predict the crash severity distributions, derived from a large number of simulations. Their results are made available in critical situations through a Random Jungle composed of 90 individual Random Forest models. The Jungle is trained with data generated using a self-developed simulation framework. A combination of a two-track dynamics model and a mass-spring model is used to simulate the future vehicle movements and evaluate the crash severity for many driving maneuvers. This is where the crash severity distribution, the Random Jungle is trained with, stems from.

It is shown for three different crash severity types that the Jungle is able to learn distribution characteristics such as the 25th and 75th percentiles. Trained with a dataset of 201,044 and tested against 74,550 distinct instances, a prediction accuracy of 85 – 98% is achieved. Finally, the Jungle is tested with data of a real dummy vehicle crash to check whether the system behavior is plausible.

Future work includes the search for solutions regarding the negative impact of delays on predictive vehicle safety systems. A further application of the CSP less prone to delays might be to estimate the crash severity for different position ns in the vehicle and use this information to relocate the passengers with the help of

highly responsive actuators. Another goal is to keep growing the FEM database to improve depending modules like the Labeler.

REFERENCES

[1] DESTATIS: Polizeilich erfasste Unfälle. (2016). [Online]. Available: https://www.destatis.de

[2] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary veh-to-veh. collision scenarios," *IEEE Trans. Intell. Transp. Sys.*, 2009.

[3] Deutscher Bundestag – Wissenschaftliche Dienste, "Vermeidung von LKW-auffahrunfällen auf bundesautobahnen durch notbremssysteme – Sachstand," September 2016.

[4] C. Grover, I. Knight, F. Okoro, I. Simmons, G. Couper, P. Massie, and B. Smith, "Automated emergency brake systems: Technical requirements, costs and benefits," TRL Published Project Report PPR 227, April 2008.

[5] T. Dirndorfer, "Integrale nutzung von pre-crash-sensorik zur ansteuerung frontaler rückhaltesys teme im fahrzeug," Doktorarbeit, Technische Universität München (TUM), Lehrstuhl für Fahrzeugtechnik, Institut für Maschinen- und Fahrzeugtechnik, 2014.

[6] M. Huang, *Vehicle Crash Mechanics*, CRC Press, 2002.

[7] L. Kübler, S. Gargallo, and K. Elsäßer, "Characterization and evaluation of frontal crash pulses with respect to occupant safety," in *Proc. 9th International Symposium and Exhibition on Sophisticated Car Occupant Safety Systems*, 2008.

[8] D. Böhmlander, V. Yano, T. Brandmeier, A. Zimmer, L. L. Ling, C. B. Wong, and T. Dirndorfer, "A novel approach for intelligent pre-crash threat assessment systems," in *Proc. IEEE 17th International Conference on Intelligent Transportation Systems*, October 2014, pp. 954-961.

[9] H. Appel, G. Krabbel, and D. Vetter, *Unfallforschung, Unfallmechanik und Unfallrekonstruktion*, Springer-Verlag, 2013.

[10] M. Botsch, *Machine Learning Techniques for Time Series Classification*, München: Cuvillier, 2009.

[11] D. Böhmländer, T. Dirndorfer, A. H. Al-Bayatti, and T. Brandmeier, "Context-aware system for pre-triggering irreversible vehicle safety actuators," *Accident Analysis & Prevention*, vol. 103, pp. 72-84, 2017.

[12] A. Meier, M. Gonter, and R. Kruse, "Precrash classification of car accidents for improved occupant safety systems," *Procedia Technology*, vol. 15, pp. 198–207, 2014

[13] M. Müller, P. Nadarajan, M. Botsch, D. Böhmländer, S. Katzenbogen, and W. Utschick, "A statistical learning approach for estimating the reliability of crash severity predictions," in *Proc. 19th Int. IEEE Conf. on Int. Transp. Sys.*, Rio de Janeiro, 2016.

[14] Federal Highway Research Institute (BASt). (2016). BASt accident research. [Online]. Available: http://www.vufo.de/forschung-und-entwicklung/gidas/?L=1

[15] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of urban mobility," *International Journal on Advances in Systems and Measurements*, 2012.

[16] L. Bieker, D. Krajzewicz, A. Morra, C. Michelacci, and F. Cartolano, "Traffic simulation for all: A real world traffic scenario from the city of Bologna," in *Modeling Mobility with Open Data*, Springer, 2015.

[17] A. Acosta, J. Espinosa, and J. Espinosa, "TraCI4Matlab," in *Modeling Mobility with Open Data. Lecture Notes in Mobility*, M. Behrisch and M. Weber, Eds., Springer, 2015.

[18] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Trans. Aeros. Elect. Sys.*, 2003.

[19] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle System Dynamics*, 1992.

[20] H. Pacejka, *Tire and Vehicle Dynamics*, Elsevier, 2005.

[21] O. Sarbaz and R. Thomson, "Influence of road characteristics on traffic safety," in *Proc. ESV 20th Conference*, 2007.

[22] I. Ahmed, "Road infrastructure and road safety," *Transp. and Comm. Bulletin for Asia and the Pacific*, vol. 83, pp. 19-25, 2013.

[23] L. Breiman, "Random forests," *Machine Learning*, 2001.

[24] F. Rosenblatt, "Principles of neurodynamics: Perceptrons and the theory of brain mechanisms," No. VG-1196-G-8, Cornell Aeronautical Lab Inc. Buffalo NY, 1961.

[25] R. Collobert and S. Bengio, "Links between perceptrons, MLPs and SVMs," in *Proc. Int'l Conf. on Machine Learning*, 2004.

[26] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

Marcus Müller received his Bachelor and Master degree in electrical engineering from Technische Universität München, München, Germany, in 2013 and 2015. Since 2015, he works as a Ph.D. candidate at the vehicle safety research center CARISSMA at Technische Hochschule Ingolstadt. As a part of the research project "Reliable Crash Prediction", which is a joint project of Technische Hochschule Ingolstadt, Technische Universität München, and AUDI AG, Mr. Müller works on the predictive recognition of traffic accidents and the time efficient estimation of the crash severity. His research interests are in vehicle safety and machine learning applications.

Michael Botsch received the diploma and doctoral degrees in electrical engineering, both with honors, from Technische Universität München, München, Germany, in 2005 and 2009. He worked for five years in the automotive industry as Development Engineer at Audi AG in the field of active safety systems. In October 2013 he was appointed Professor for Vehicle Safety and Signal Processing at Technische Hochschule Ingolstadt in the Department of Electrical Engineering and Computer Science. He is the Associate Scientific Director of the vehicle safety research center CARISSMA at Technische Hochschule Ingolstadt. His research interests are in signal processing and automotive applications. Prof. Botsch is a member of IEEE and VDE.

Dennis Böhmländer is working as a development engineer for incident detection at AUDI AG, Ingolstadt. Until August 2015, he was a research assistant with the Institute of Applied Research of Technische Hochschule Ingolstadt. In 2017 he has received his PhD degree titled "Innovative Crash-sensing Architectures - A new approach in contact-less vehicle crash detection" in electrical engineering from the De Montfort University, Leicester, UK. His main research interests include intelligent transportation, crash data analysis, predictive algorithms and smart technologies for crash detection. The goal of his research is to improve the safety of transportation systems.

Wolfgang Utschick's (b. 1964) research combines different areas of applied mathematics in various signal processing application domains as in the field of wireless communications, radar and automotive safety. Most recently he also started research projects in the area of power transmission grids. He holds several patents in the field of signal processing and has authored and co-authored a great many of technical articles in international journals and conference proceedings.
Prof. Utschick studied and did his doctorate at TUM after working in the industry. After a short postdoctoral period at TUM and ETH Zürich, he was appointed to the Professorship for Signal Processing Methods in 2002. Wolfgang Utschick has been principal investigator in multiple collaborative projects with the industry and in many research projects funded by the German Research Foundation (DFG). Since 2011 he is member of the steering committee of the TUM Department for Electrical and Computer Engineering (ECE), first as the Dean for Study Affairs and since 2017 as the Dean of the department.