AI for Safety: How to use Explainable Machine Learning Approaches for Safety Analyses

Iwo Kurzidem[1,*], Simon Burton[1] and Philipp Schleiss[1]

[1]Fraunhofer Institute for Cognitive Systems IKS, Hansastraße 32, D-80686 Munich

Abstract

Current research in machine learning (ML) and safety focuses on safety assurance *of* ML. We, however, show how to interpret results of explainable ML approaches *for* safety. We investigate how individual evaluation of data clusters in specific explainable, outside-model estimators can be analyzed to identify insufficiencies at different levels, such as (1) input feature, (2) data or (3) the ML model itself. Additionally, we link our finding to required artifacts of safety within the automotive domain, such as *unknown unknowns* from ISO 21448 or *equivalence class* as mentioned in ISO/TR 4804. In our case study we analyze and evaluate the results from an explainable, outside-model estimator (i.e., white-box model) by performance evaluation, decision tree visualization, data distribution and input feature correlation. As explainability is key for safety analyses, the utilized model is a random forest, with extensions via boosting and multi-output regression. The model training is based on an introspective data set, optimized for reliable safety estimation. Our results show that technical limitations can be identified via homogeneous data clusters and assigned to a corresponding equivalence class. For unknown unknowns, each level of insufficiency (input, data and model) must be analyzed separately and systematically narrowed down by process of elimination. In our case study we identify "Fog density" as an unknown unknown input feature for the introspective model.

Keywords

safety analysis, safety engineering, explainable machine learning, outside-model estimator, safety validation

1. Introduction

The use of artificial intelligence (AI) and especially machine learning (ML) in safety critical applications, such as autonomous driving (AD), is still a vivid research area, as many state-of-the-art ML methodologies create end-to-end trained (i.e., black-box) models for object detection and localization [1]. Encoded into these black-box models are performance and specification insufficiencies that cause epistemic and/or aleatoric uncertainties [2]. Identifying, estimating and, if possible, mitigating uncertainties is required for a convincing safety assurance [3]. Figure 1 provides an overview of different uncertainty manifestations typical for ML:

- Input feature: Is the ML model's decision process based on the correct input factors from the complex environment?
- Data: Does the collected data (training & test) include enough and proper samples with an appropriate distribution?
- ML model: Is the selected ML methodology appropriate for the desired task?

Finding and understanding the root cause(s) of uncertainty and identify the corresponding insufficiency is not a trivial task, as typical results from quantitative tests of the ML model do not allow a straightforward mapping between a measured lack of performance to a specific insufficiency, due to complex interdependence and correlations between the causes.

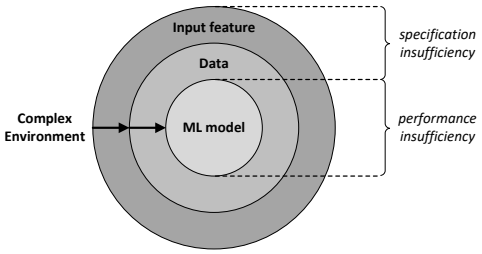

Figure 1: Uncertainties associated with ML. Adapted from [3].

The main contribution of this paper is an approach to identify specific insufficiencies and eventually link the analysis results to required artifacts of automotive safety standards, for example related to *unknown unknowns* from ISO 21448 - Safety of the intended functionality (SOTIF) [2] or *equivalence class* for validation from ISO/TR 4804 [4]. In doing so, we present a solution to address open issues in ML safety assurance regarding safety tests, such as how many tests have to be performed within which operational design domain (ODD) [5].

In previous work we presented a conceptual framework to create an explainable, introspective model (i.e., white-box) from a deep neural network (i.e., black-box), cf. Fig 2. In a case study, we used the approach to estimate

The IJCAI-2023 AISafety and SafeRL Joint Workshop
*Corresponding author.
iwo.kurzidem@iks.fraunhofer.de (I. Kurzidem);
simon.burton@iks.fraunhofer.de (S. Burton);
philipp.schleiss@iks.fraunhofer.de (P. Schleiss)
0000-0001-9040-8752 (S. Burton)

Figure 2: From Black-box to White-box. Adapted from [6].

the safety and reliability of the black-box via the white-box for object detection in the automotive domain. While the developed white-box models showed some promising results, such as providing estimated distributions for successful and failed defections, their unrestricted usage for safety assessment is currently not possible, details see [6]. In this contribution we use the developed models *for* safety analyses to identify specific insufficiencies. We leverage the fact that random forests (RFs) contain interpretable decision trees (DTs) and analyze the obtained DTs with regards to split criteria and data clustering.

This paper is organized as follows. Section 2 provides an overview of relevant and related works. We continue by introducing our approach and its basic premise in Section 3. Next, in Section 4, we demonstrate our approach and perform corresponding analyses. Finally, in Section 5, we conclude the paper by summarizing our results and discussing future work.

2. Related Works

Currently, most research on AI for AD focuses on improving the safety related aspects of ML models itself. Either by means of conventional (i.e., non-ML) analysis methods [7] or methods directly enhancing the ML model [8]. These conventional safety methods include hazard and risk analysis [9], simulation [10, 11], (stochastic) fault tree analysis [12] etc., while ML specific methods for safety cover uncertainty quantification [8] and robustification [13] among others. However, conventional safety methods are not particularly well suited for safety considerations regarding AI, such as the definition of equivalence classes of safe or unsafe behavior or discovering unknown unknowns, as these characteristics manifest themselves differently in ML-based systems, due to correlation of input to output instead of causality of data processing. Enhancing ML models requires modification of the base network, without providing traceable safety artifacts. Therefore, new safety analysis methods are needed, including approaches leveraging ML itself. Similar to [14], which uses a Bayesian network to identify novel triggering conditions, as required by SOTIF.

The German Federal Ministry for Economic Affairs and Climate Protection initiated the project "KI-Absicherung" (KI-A), consisting of 24 partners from industry and academia, to address the complex topic of AI and safety in the mobility market [15]. The main focus of KI-A was the development of a methodology for safety assurance for ML algorithms, in particular for object detection and instance segmentation. Most of the used approaches for safety included conventional methods, such as visual analytics [16], combinatorial testing [17], data augmentation [13] and others. All these methods work within a well defined, limited semantic space. A couple of methods in KI-A used ML techniques, such as principal component analysis (PCA) [15] and search-based testing [18], to specifically analyze and search for insufficiencies in data. However, all of these methods require some insights or a-priori knowledge about the root cause of the specific insufficiency to be applied successfully. Our approach does not assume any specific insufficiency from the outset, instead each layer of uncertainty (cf. Fig. 1) is analyzed by itself and by process of elimination the root cause is identified.

Besides KI-A and beyond AD, ML has successfully been used for data clustering and analysis, such as PCA, k-means or Latin hypercube sampling, to define relevant sceneries to reduce the effort of verification and validation [19]. Again, none of the mentioned methods explores all the different possible insufficiencies due to input, data or model, but instead already know where to look.

3. Methodology

In [6] we introduced a framework to create explainable, introspective white-box models, derived from black-box model test evaluation, to predict different safety related aspects of the deep neural network (DNN) object detector. Unfortunately, the measured performance of the white-box models did not allow for an unrestricted use as reliable safety monitors. In this contribution we investigate if we can use the white-box models themselves to analyze certain safety properties and link the obtained result to insufficiencies within different layers, cf. Fig. 1. Put differently, can we use the semantic input of the white-box to characterize the black-box regarding safe, unsafe and unknown behavior.

On the one hand, we examine the single DTs of the RF white-box models to identify possible equivalence classes. This enables us to possibly define an efficient test strategy for verification and validation further down the ML development-cycle. On the other hand, we investigate if contradictory samples within DT leafs indicate unknown unknowns. Here unknown unknowns represent previously unconsidered parameter from the complex environment, not part of the initial problem space.

Regarding results, the analysis of DT leafs might *not* end conclusively for either equivalence classes or unknown unknowns. This does not mean there are definitely no such cases to be found, but instead that, given the input space, equivalence classes or unknown un-

knowns are unlikely to be found within these data.

In principle the proposed approach can be applied to *any* kind of ML data, however, it greatly benefits from certain restrictions to be usable in safety. Firstly, the input dimensions should have a semantic description, meaning they have an humanly interpretable representation in the real world. For instance, a semantic dimension may refer to an object's attribute (e.g. size) or environmental conditions (e.g. rain), whereas non-semantic descriptions include technical aspects (such as pixel intensity, blur, etc.). Secondly, the input space should be limited. The aggregation and interpretation of multiple and different input parameters may result in too complex cases to be analyzed and used in safety argumentation.

The basic concept of DTs is data partitioning [20]. To this end, the input space of data is repeatedly partitioned into disjoint, smaller subsets, such that each subset is consistent with regards to the desired output. A visualization of a simple DT is given in Fig. 3. As can be seen, the input data is partitioned into subsets by splitting at each node, using the most suitable input feature (in conjunction with a specified error function, details in section 3.1). The final data clusters, i.e., the leafs of DTs (from now, we will use the terms interchangeably) represent the "most consistent" partitioning given the defined hyperparameters and provided data. The collection of multiple DTs together is RF and this ensemble provides its final output by aggregating the prediction of each single DT. There are different versions of RFs, such as bagging and boosting extensions, that differ in way the DTs are created from the provided data (see section 4.2). The mathematical fundamentals to create DTs, such suitable split criteria s, and their interpretation for safety analyses are given in following sections.

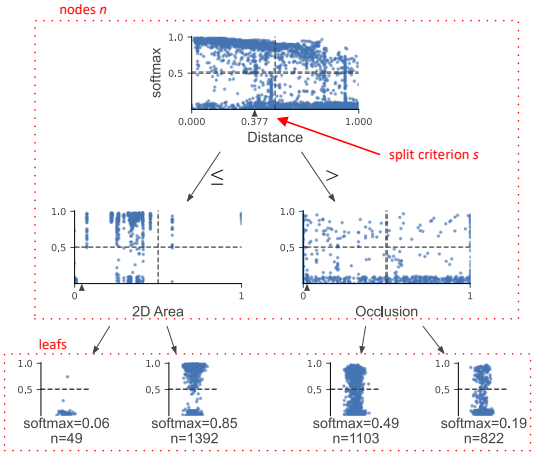

Figure 3: Simple decision tree (DT) with visualized data partitions.

3.1. Equivalence classes of equal behavior

The underlying methodology of DTs creates disjoint subsets of inputs that produce the same output (while minimizing variance) [21]. This is very similar to the definition of equivalence class from ISO/TR 4804 [4], which states that, equivalence classes are based on the division of inputs and outputs, such that a (single) representative test can be defined. Therefore we use the leafs of DTs to define an equivalence class. In addition, we use the quantitative split criteria $\{s_1, ..., s_n\}$ of the DTs node's, to define the boundaries (i.e., limits) of the corresponding equivalence class, cf. Fig. 3.

The foundation of DTs is data partitioning by (binary) splits, to uncover complex patterns. For each possible binary split value s at node n the resulting *decrease in impurity* $\Delta(s, n)$ is being determined by [21]:

$$\Delta(s,n) = f_i(n) - \frac{S_{nL}}{S_n} * f_i(nL) - \frac{S_{nR}}{S_n} * f_i(nR), \quad (1)$$

with S_n denoting the size of the training data for node n, S_{nL} and S_{nR} representing the samples from the whole training data assigned to the left child and right child respectively, and f_i as the impurity function. The maximization of decrease in impurity can be understood as best possible split s for node n into two children (nL and nR). For regression tasks, typically the *squared error loss* is being computed with Eq. (1), to determine the error during training. Therefore, $f_i(n)$ calculates the local, i.e. for a specific node n, squared error loss via [21]:

$$f_i(n) = \frac{1}{S_n} \sum_{x,y \in L_n} (y_M - y_T)^2. \quad (2)$$

In Eq. (2), x denotes a specific input feature and y the corresponding model output from the subset of learning samples L_n. y_M and y_T are the model output and desired output respectively. Both equations, (1) and (2), essentially split the data into clusters that produce the most similar output. Figure 4 shows an example for data splitting, containing measurement samples for object distance (input feature x) and corresponding softmax confidence (y_T). The best split s divides L_n into two clusters, S_{nL} and S_{nR}, that have the highest decrease in impurity. The horizontal lines within the left (S_{nL}) and right cluster (S_{nR}) indicate the arithmetic mean for each of them. Any other split, for instance s^* (cf. Fig. 4), yields:

$$\Delta(s,n) > \Delta(s^*,n). \quad (3)$$

Using DTs and input features with semantic meaning, that can be measured quantitatively, all splits $\{s_1, ..., s_n\}$ along one path, from upper parent to lower child, define the limits of a potential equivalence class.

It is important to note, that simply aggregating all split values $\{s_1, ..., s_n\}$ along one branch within a DT *does*

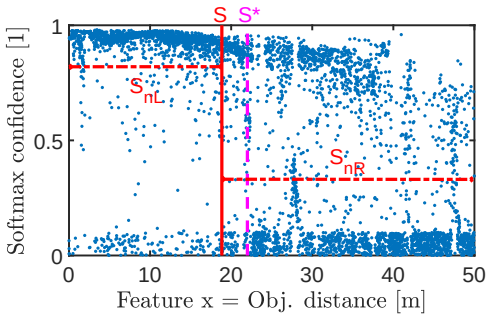

Figure 4: Example of split s partitioning the data into S_{nL} and S_{nR}.

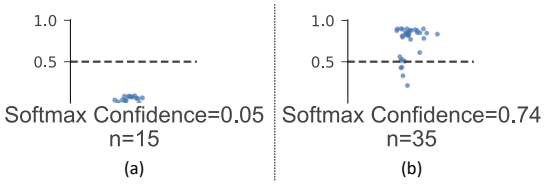

Figure 5: (a) Leaf that shows a potential equivalence class, (b) Leaf that contains inconsistent data points.

not guarantee an equivalence class. The methodology of RFs and DTs requires some hyperparameters to be set that influence the splitting and, therefore, the resulting clusters. Most important for our considerations are:

- Threshold θ for the minimum decrease in impurity, i.e., $\Delta(s, n) < \theta$,
- The minimum amount of samples S_{min} to allow further splits, i.e., $S_n > S_{min}$.

The first threshold θ prevents an overfitting, as no threshold allows the splitting of virtually identical values as long as there is *any* decrease in impurity. Refer to Fig. 4, nearly all measured softmax confidence values will be different after r decimal places (dependent on the precision of the data). Therefore, even splitting samples that vary after r digits will decrease impurity, eventually creating DTs with one single data point per leaf. The second parameter S_{min} also prevents overfitting. Lets assume that S_{min} is set to the smallest possible value, which is 2. Given a small enough θ, each single leaf will converge at single data points. Therefore, both, θ and S_{min} together, influence the resulting clusters and if meaningful equivalence classes can be defined. Please note, that there are more hyperparameters to prevent overfitting, but they are not relevant for this contribution. Please also note, during our analyses (Section 4) we did inspect all of the possible hyperparameters that could in principle provide an explanation for the seen results, e.g. `tree_depth`, to make sure they are not responsible for it.

In order to define an equivalence class, the DT leaf *must* contain more samples than S_{min}, i.e., $S_n > S_{min}$. The basic reasoning is the following, if a leaf contains more samples than S_{min} a split could have been possible, however, it was *not necessary* as θ has not been exceeded. To put it differently, there are no more disjoint subsets within these data, cf. Fig. 5(a). The only other possibility is that a further split was *not possible* although θ allowed for it, given the model, data and input features. Such a leaf can indicate unknown unknowns, cf. Fig. 5(b).

3.2. Unknown unknowns

The goal of SOTIF is to identify potential unknown hazardous scenarios, arising from the interaction between the system and its complex environment, and mitigate their effects. To archive this, SOTIF recommends to search for triggering conditions that lead to potential hazardous scenarios. Unfortunately, there is no established approach or method to identify such triggering conditions for all possible systems and environments. Furthermore, the nature of some of these triggering conditions can be defined as unknown unknowns, i.e., something we are not even aware that we do not know. In our context it refers to a feature of the input space that was not considered when approximating the factors that influence the performance of the black-model.

The key idea is to identify and use inconclusive, yet interpretable data clusters and, by process of elimination, show that the only possible explanation for their existence is an unknown unknown. In the previous section 3.1, we examined the mathematical foundation for data clustering via DTs. In particular equations (1) and (2) partition the available data into the best possible disjoint and coherent clusters. However, in some cases the resulting, final clusters still have high impurity, although further splitting, in principle, is allowed. Simply put, the cluster contains contradictory data, which cannot be split meaningfully anymore within the defined scope, cf. Fig. 5(b).

How can this be interpreted? Given that the hyperparameters θ and S_{min} are not exceeded, either the input, data or model did not allow for any further optimization. Now each single layer (cf. Fig. 1) and potential insufficiency must be analyzed on its own to identify the root cause. To clearly uncover an unknown unknown, neither data nor model shall be the root cause of the impure data clustering. Only if a "seemingly" new *input feature* can resolve the contradiction, a unknown unknown is plausible. "Seemingly", as it is yet unknown, even by the process of elimination, if such a semantic feature can be identified and if so, which one it is specifically. Regarding the modelling, only explainable or interpretable models are useful for the presented approach, as only those allow to define comprehensible equivalence classes. To

investigate whether the modelling itself is responsible for the inhomogeneous clustering of data, alternative or modified approaches for model y_M should be deployed and compared. For data, the corresponding distributions of the input features $\{x_1, ..., x_j\}$ within the boundaries of the potential unknown unknown must be investigated.

Do note, that there are numerous leafs per DT that are endpoints due to the thresholds of θ or S_{min} being reached. These clusters cannot be interpreted as neither equivalence class nor unknown unknowns. Remember that θ and S_{min} primarily prevent overfitting. On the one hand, smaller and smaller values for θ and S_{min} will converge on clusters with single data points. Consequently, creating equivalence classes which are correct from a safety point of view, but carry no useful information. On the other hand, larger values will always serve as limits for the clusters, and it is *impossible to know* if additional clusters where not necessary or not possible, and as such offer no information about potential unknown unknowns.

4. Safety Analyses

Based on the results from [6], we conduct our safety analyses and demonstrate the presented methodology via a case study. In our previous contribution we recognized that the reliability of the RFs models is not sufficient for an unrestricted usage for safety. Therefore, we specifically analyzed the model $RF_{softmax}$ regarding its single DTs, including their split criteria and leaf clusters, to explain the mixed performance results. $RF_{softmax}$ estimates the reliability of the provided softmax confidence from a DNN object detector. Please note, that in order to use model $RF_{softmax}$ as a safety predictor, specific input features from the complex environment, which are arguably safety-relevant, have been pre-selected.

To create $RF_{softmax}$ the implementation from *scikit-learn* was used, with thresholds $S_{min} = 10$ and $\theta = 0.001$. For other hyperparameters, please refer to [6]. An investigation of $RF_{softmax}$ revealed strong similarities between the single DTs within the model. Additionally, the DTs occasionally expressed leafs cluster similar to the ones shown in Fig. 5(a) and (b). A further analysis of all leafs from the DTs revealed three basic cases:

1. Leafs that show *little variance* in data and fulfill $S_n = S_{min}$,
2. Leafs that show *little variance* in data and fulfill $S_n > S_{min}$,
3. Leafs that show *high variance* in data and fulfill $S_n > S_{min}$.

The first case is the most common one. According to equations (1) and (2), together with a suitable θ and S_{min}, the RF methodology created the best possible leafs, while preventing overfitting. These clusters represent a reasonable model, but no useful information for safety can be extracted. The second case is an interesting abnormality, as it signifies an early stopping. Given θ, it was not necessary to create additional child clusters, as the decrease in impurity is insignificant. In brief, all data expressed the same output behavior without colliding with the hyperparameter thresholds. This case will be discussed in detail in Section 4.1. The last case shows impure clusters, although the defined hyperparameters did not account for this. Therefore, the root cause for this inhomogeneous data must lie within one of the different layers, as shown in Fig. 1. This is the object of Section 4.2.

With these analyses we try to identify insufficiencies and link our finding to safety artifacts from ISO 21448 and ISO/TR 4804.

4.1. Equivalence class of equal DNN behavior

Following the identification of the three basic cases, the most promising leafs for both, overall $RF_{softmax}$ performance and safety significance, are leafs that accumulate many similar data points without surpassing any of the defined limits of the hyperparameters θ and S_{min}. Therefore, if $S_n > S_{min}$ is true, at least *one* input feature x is a coherent predictor. These clusters can be identified by searching the final number of samples per leaf and comparing them to S_{min}.

The methodology of RFs creates each DT from a subset of the complete training data. Therefore, all DTs are based on slightly different data sets and identified, potential equivalence classes may only exist within one single DT and not represent an overall equivalence class. In order to to verify a potential equivalence class, the aggregated split criteria $\{s_1, ..., s_n\}$ should be applied to the complete data set. If all the samples show a similar output, an equivalence class can, in principle, be defined. For our presented analysis we selected the most promising equivalence class, i.e. the least restrictive one regarding its split criteria $\{s_1, ..., s_n\}$, out of all potential candidates. Table 1 shows an identified equivalence class that also represents a technical limitation of the trained black-box object detector. All objects with an detection area smaller than $3.6233m^2$, at a noise level of at least of 74%, do not

Table 1
Example of an identified equivalence class.

Input feature x	Interval s_n	Unit
Object distance	all	[m]
Object area	$x \leq 3.6233$	[m^2]
Object occlusion	all	[%]
Noise variance	$74 \leq x$	[%]

have a softmax confidence higher than 0.1, cf. Fig. 6. Effectively, none of such objects are being detected by the black-box object detector, independent of distance or occlusion. In terms of ISO/TR 4804 equivalence class, this means, that for all samples fulfilling Table 1 one singular test is sufficient to verify the black-box system's response.

Apart from such a successful equivalence class, some of the potential clusters do not exhibit the same behavior over all corresponding samples. The split criteria $\{s_1, ..., s_n\}$ do not represent an equivalence class, if they are only true within specific DTs, but not for the complete data. Figure 5 shows a verification of two potential equivalence classes. The first plot (eq. class) visualizes the softmax confidence for all samples complying to Table 1. This equivalence class has been derived from multiple DTs, on average with $S_n = 15$. In contrast, an example of a plot (invalid cluster) for a potential equivalence class that is not homogeneous for all samples within the identified $\{s_1, ..., s_n\}$.

Besides the verification via sample outliers, the equivalence classes that showed homogeneous output in all data have also been "qualitatively" verified by testing the black-box detector. In this context qualitatively means, that the simulation environment of CARLA [22] does not allow a specific object size to be set, instead predefined assets can be selected and deployed, however, the precise object area (within a frame) needs to be derived and transformed (incl. rounding and translation errors) to fit the developed safety framework [23]. Therefore, the *exact* object area of 3.6233m^2 as limit could not be verified beyond any doubt.

For the equivalence class provided by Table 1 a set of test cases have been created. One such scenario, with multiple objects and detection areas smaller and larger than ~3.6233m^2, has been generated and tested, cf. Fig. 7. Indeed, the verification result of the different test cases confirm this combination of noise variance and object area as credible detection limit. However, the verification also revealed that this equivalence class represent the *upper* (or *lower*) limit. For instance, objects are sometimes lost before the limits have been reached. Within this contribution we did not investigate, whether these results could be used to refine the limits of the identified equivalence class into fine-grained subcategories (cf. Table 1). Especially, since transformation and translation errors could not be ruled out entirely.

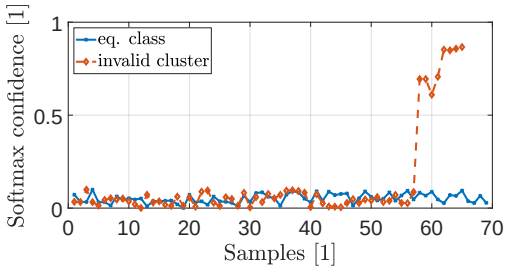

Figure 6: Examples of (un)successful equivalence classes.

Figure 7: (left) Detection of multiple objects under ideal conditions. (right) The noise level has been increased to 74%, only the object with area 3.753m^2 is still detected.

During the safety analysis to positively identify equivalence classes, almost all of results converged on a combination of factors representing a technical limitation of the system. Such as robustness against noise and area of the object or maximum detection distance. The remaining cases that are seemingly not technical limitations, but do show convergence, are still under investigation regarding their meaning (as they require very accurate CARLA simulation and transformation).

Due to the abstraction of the input space by the methodology of [6], the identified equivalence class can be used as *logical scenario*, see [24], for ISO/TR 4804 validation efforts.

4.2. Unknown unknowns (of white-box)

Another anomaly within the DT structure are leafs that show high variance in data, but seem to not gain anything from additional splits, i.e., $S_n > S_{min}$. Equations (1) and (2) ensure that the best possible data clusters are being created, except if this is impossible, given either model, data or input. One such instance is shown in Fig. 5(b). Regarding this cluster, we selected it specifically, as it appears to be most suitable, due to its comparatively broad limits $\{s_1, ..., s_n\}$ for the input features. Similar leafs has been identified as reoccurring pattern across multiple DTs. After aggregation of split criteria, the leafs in question converge on the criteria presented in Table 2. The appearance of such clusters is one explanation for the mixed performance results of the model $RF_{softmax}$, as reported in [6].

According to Fig. 1, the first layer to investigate a performance insufficiency is the ML model itself. In order to determine if the modelling approach itself is responsible for this, modified approaches have been implemented and analyzed. Specifically, we used the RF extensions of boosting (via *LightGBM*) and multi-output regression (via *XGBoost*) for python. Boosting (by weighing) uses a combination of bootstrap and evaluated test data to train

Table 2
Inhomogeneous cluster of a DT and its boundaries.

Input feature x	Interval s_n	Unit
Object distance	$18.85 \leq x \leq 31.25$	[m]
Object area	$2.018 \leq x$	[m^2]
Object occlusion	all	[%]
Noise variance	$62 \leq x \leq 78$	[%]

the successive DT [25]. The idea is, that this methodology explicitly tackles high variance leafs, as it penalizes misclassification by weighing the entire training set L_n accordingly. With multi-output regression, several output variables are simultaneously predicted [21]. In [6] we trained three different models for three different target variables. Via multi-output regression we hope to leverage some dependencies between these output variables, such as a correlation between softmax confidence and bounding box size shifts. The minimization of impurity, Eq. (1), via the squared error (2) is fundamental to all of the extensions. Please remember, the selection of suitable approaches is limited by the necessity for explainability.

The evaluation of the overall performance for all models reveals that the measured performance converges, see Fig. 8. All three models display a relatively high amount of correct predictions for very low and high softmax confidences. Be aware, that the model `Multi-output` has a slightly smaller test set, as for its sequential models building process the samples with false negatives cannot be used. In terms of quantitative values, the Mean Squared Error (MSE) and Mean Absolute Error (MAE) show maximum improvements of $\Delta MSE \leq 1.22e^{-2}$ and $\Delta MAE \leq 2.32e^{-2}$ between the new models and `RF base` (with MSE= $2.25e^{-2}$ and MAE= $8.00e^{-2}$). Unfortunately, this means no model performs significantly better than the others. Due to the different training approaches between the models, a detailed comparison of leafs and structure is not possible without extensive effort. This result either indicate, that these kind of models are inherently unable to predict the black-box behavior or that there are specification insufficiencies in data and/or input that cause this response. The outcome of all of this is, changing the model does not seem to resolve inhomogeneous clustering, as outliers are apparent for all models (cf. Fig. 8). On account of this, we continue by investigating possible unknown unknowns by visualizing the relevant data distribution.

By the process of elimination, to rule out implausible root causes, we arrive at the collected data. We continue by highlighting the data distribution given by Table 2. Figure 9 displays the data points for the relevant input features of Obj. distance and Noise variance, narrowed down by the specific split criteria $\{s_1, ..., s_4\}$. For a convenient visualization, the data points of $2.018m^2 <$ Obj. area have been filtered out. Also, the corresponding softmax score is divided into low and high. One distinctive feature of Fig. 9 is the relative high amount of data points that show high and low confidence at the same time for Obj. distances of around 21 m. This contradiction can seemingly not be resolved by recruiting additional input features, such as Obj. occlusion. The existence of such data points provides one plausible explanation why the cluster is inhomogeneous, despite $S_n > S_{min}$. Additionally, there exists an almost straight line of low confidence scores at 23 m. This is most likely indicating a technical limit, but as this cluster could not be split further by available input dimensions, it must not be represented well in the available data. On the whole the displayed section, limited by Table 2, could not be split into homogeneous clusters by any of the available input dimensions.

Taking into consideration the distribution of Fig. 5(b), more data samples will most likely *not* enforce another split into more homogeneous clusters, as $S_n > S_{min}$ already indicates that this is not the root-cause. The only case where additional samples help, is if the underlying data distribution within the *other* input features are not appropriate, i.e. imbalanced, as this represents skewed information. An investigation of data revealed, that the distributions for object occlusion and object area are not entirely balanced. The reasons are, objects have fixed sizes and for occlusion at least two objects are required, one of which is definitely not occluded, while with three or more objects multiple ones are fully occluded, thus creating small biases. However, all in all the data distribution is considered sufficiently good to rule it out as root cause for the poor DT data clustering.

If great data imbalance is not evident, there are only two possible impacts additional samples can have. Either, (a) extra measurement samples skew the distribution into a certain direction (basically creating a bias), but still, the

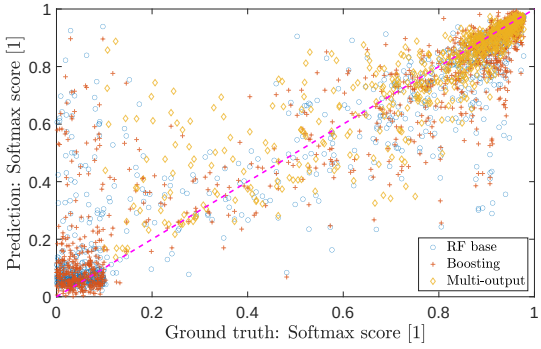

Figure 8: Different explainable models and their performance (diagonal line shows ideal behavior). None of them shows a definitive advantage over the others, suggesting a root cause independent of the selected ML methodology.

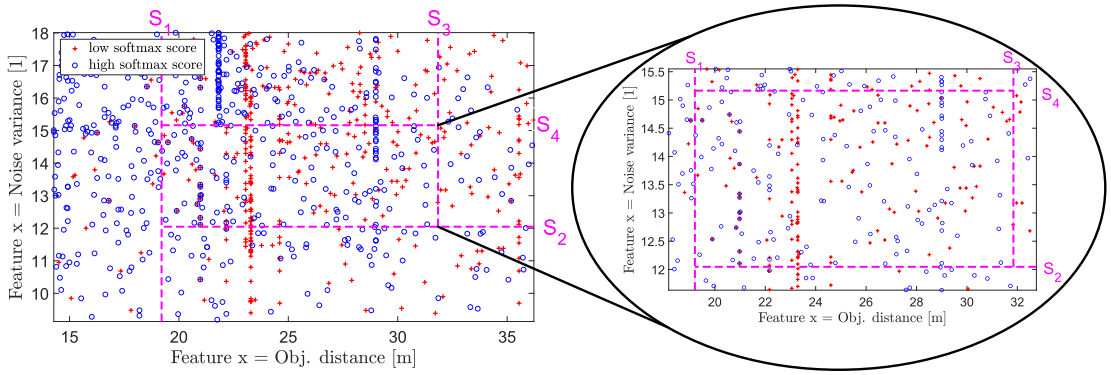

Figure 9: (left) Data distribution for features Obj. distance and Noise variance. Red crosses denote low softmax confidence (softmax < 0.5), while blue circles show high softmax confidence (softmax ≥ 0.5). (right) Zoom in of the data under investigation (cf. Table 2).

cluster would remain collectively inhomogeneous, or, (b) the new samples *alone* can be partitioned into its own cluster (split by the remaining, available input features). Although investigating cases (a) and (b) could provide additional information, no experiments have been carried out within this contribution, as the expected results would not impact the next analysis.

All the previous analyses lead us to the only plausible conclusion: The introspective data set does not include all the necessary data dimensions.

The next step involves reviewing the input features x. In order to act as an explainable, introspective model, the input space for the white-box model has been reduced to certain input features, called safety features, in [6]. Following the process of elimination, neither the model nor the data provide any convincing evidence that they cause this observed inconsistency, cf. Fig. 5(b). Therefore, only the input features remain as possible root cause. The input features in [6] have carefully been selected based on two principles that ensure safety-relevance and redundancy:

1. The feature is safety-relevant, i.e., factors that typically cause traffic accidents in human driving,
2. The feature must be measurable via a different sensor, i.e., independent of the black-box prediction.

These principles still apply, so possible new features must adhere to these principles to be useful for a reliable safety monitor.

The basic strategy to discover possibly new, important input features revolves around the idea to use the evaluated analysis results from the previous tests. According to the split criteria of Table 2, occlusion effects are unimportant and object's area only requires a minimum value for detection (given the noise level interval). Therefore, the new input feature should not correlate with either of these, as they do not carry any useful information to disentangle the data, cf. Fig. 9. We also excluded biases.

During our initial inspection of the data we already identified one irregularity, namely, data points that have low softmax confidence at a specific Obj. distance x = 23m across virtually all noise levels. Although this is not completely uncommon, see areas outside the highlighted cluster in Fig. 9(left), in this particular case, however, *none* of the other input features could be recruited to separate these outliers. Subsequently, we examined the corresponding frames in order to determine a potential effect that could cause such a hard limit.

Our review revealed, that for all these frames the `carla.WeatherParameters` contained a nonzero value for `fog_density`. In our initial setup to create an explainable, introspective model we introduced "Noise variance" as technical implementation for *all* environmental disturbances, such as rain, fog or white-noise. So these effects cannot be distinguished from each other within the introspective data set. As a result, they act as unknown unknowns within this system's environment (introspective model). Although rain and fog produce similar visual effects in CARLA, fog acts as a limitation for the maximum field-of-view distance and therefore also limits the capabilities of the black-box object detector. With regards to the (safety) principles 1. and 2., the feature "Fog density" can definitely be classified as safety-relevant and also be detected via other sensors. A linear correlation analysis has been carried out to determine the dependence of Fog density with other input features, see Fig. 10. As this table shows, a strong positive correlation exists between noise and fog, as the basic simulation effect is similar. It can also be seen, that both, noise and fog, show minimal correlations with the other, remaining input features. This indicates a good candidate for a *now*

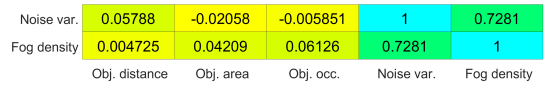

Figure 10: Correlation heatmap for the features Noise variance and Fog density of the the introspective data.

(un)known unknown.

Based on these findings, we separated "Fog density" from Noise variance and included the meta-data in the introspective data set as new input feature. The preliminary experiments indeed show an improvement. Since a new input feature was introduced, the resulting DTs cannot simply be compared. It is, however, possible to filter for all the leafs that fall within the previous boundaries of Table 2. This inspection showed that additional, improved sub-clusters have been created, see Fig. 11. By identifying and including a previously unknown unknown input feature, the previously inconsistent data cluster could successfully be subdivided into more balanced leafs, showing the relevance of this input dimension for the introspective model. Please be aware that the new sub-clusters can still result in *any* of the three basic cases for DT leafs (cf. Section 4), so the analysis might not end conclusively every time.

5. Conclusion and Future Work

The work presented in this paper shows how explainable ML can help and guide us to discover equivalence classes (ISO/TR 4804) and unknown unknowns (SOTIF). The developed approach makes use of the mathematical foundation of DTs to identify leafs and interpret their meaning, with respect to defined thresholds and their degree of data variance. We successfully use the methodology to define an equivalence class (Table 1) and uncover an unknown unknown (Fig. 11) for the application of a explainable, outside-model estimator.

Some question, however, do remain. While some equivalence classes can be identified and meaningfully interpreted, other cases beyond system (capability) limitations are difficult to humanly comprehend. Within the described use case we were able to identify *one* unknown unknown by disentangling *one* inconsistent data cluster.

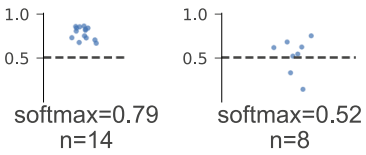

Figure 11: One set of sub-leafs for cluster Fig. 5(b), after introduction of feature "Fog density".

With typically multiple such data clusters this approach might not scale particularly well. Additionally, the introduction of another input feature requires the reevaluation of the previously identified equivalence class. In our case input feature "Noise variance" was *changed after* the successful definition of the equivalence class. Besides, this work benefits from the already limited input space from the introspective model, identifying unknown unknowns is probably not as straightforward for other use cases, especially if many options are available. Moreover, the defined problem space, in our case the reliability of the softmax confidence, also defines the domain for potential unknown unknowns. Other unknown unknowns, not related to this model, may remain hidden and represent a remaining residual risk that must be quantified beyond our model. These aspects will be part of future research for this approach.

Acknowledgments

This work was funded by the Bavarian Ministry for Economic Affairs, Regional Development and Energy as part of a project to support the thematic development of the Institute for Cognitive Systems.

References

[1] E. Yurtsever, J. Lambert, A. Carballo, K. Takeda, A survey of autonomous driving: Common practices and emerging technologies, in: Proc. IEEE Access, volume 8, 2020, pp. 58443–58469.

[2] International Organization for Standardization, Safety Of The Intended Functionality - SOTIF (ISO/-PAS 21448), ISO, 2019.

[3] S. Burton, B. Herd, Addressing uncertainty in the safety assurance of machine-learning, Frontiers in Computer Science Hypothesis and theory article (2023).

[4] International Organization for Standardization, Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation (ISO/TR 4804:2020), 2020.

[5] P. Schleiss, Y. Hagiwara, I. Kurzidem, Towards the Quantitative Verification of Deep Learning for Safe Perception, in: Proc. 2022 IEEE Int. Symp. on Software Reliability Engineering Workshops (ISSREW), 2022, pp. 208–215.

[6] I. Kurzidem, A. Misik, P. Schleiss, S. Burton, Safety Assessment: From Black-Box to White-Box, in: Proc. 2022 IEEE Int. Symp. on Software Reliability Engineering Workshops (ISSREW), 2023, pp. 295–300.

[7] S. Burton, I. Kurzidem, A. Schwaiger, P. Schleiss, M. Unterreiner, T. Graeber, P. Becker, Safety As-

surance of Machine Learning for Chassis Control Functions, in: Proc. Int. Conf. on Comp. Safety, Reliability, and Security, Cham, 2021, pp. 149–16.

[8] A. Schwaiger, P. Sinhamahapatra, J. Gansloser, K. Roscher, Is Uncertainty Quantification in Deep Learning Sufficient for Out-of-Distribution Detection?, in: Proc. AISafety@IJCAI, 2020.

[9] S. Khastgir, H. Sivencrona, G. Dhadyalla, P. Billing, S. Birrell, P. Jennings, Introducing ASIL inspired dynamic tactical safety decision framework for automated vehicles, in: Proc. 2017 IEEE 20th Int. Conf. on Intelligent Transportation Systems (ITSC), 2017, pp. 1–6.

[10] P. Koopman, M. Wagner, Toward a Framework for Highly Automated Vehicle Safety Validation, Technical Report, SAE Technical Paper, 2018.

[11] D. Rao, P. Pathrose, F. Huening, J. Sid, An approach for validating safety of perception software in autonomous driving systems, in: Proc. Model-Based Safety and Assessment: 6th Int. Symp., IMBSA 2019, Thessaloniki, Greece, October 16–18, 2019, Proc. 6, 2019, pp. 303–316.

[12] M. Ghadhab, S. Junges, J.-P. Katoen, M. Kuntz, M. Volk, Model-based safety analysis for vehicle guidance systems, in: Proc. Comp. Safety, Reliability, and Security: 36th Int. Conf., SAFECOMP, 2017, pp. 3–19.

[13] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, B. Lakshminarayanan, AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty, arXiv:1912.02781 [cs, stat] (2020). arXiv:1912.02781.

[14] A. Adee, R. Gansch, P. Liggesmeyer, C. Glaeser, F. Drews, Discovery of Perception Performance Limiting Triggering Conditions in Automated Driving, in: Proc. 2021 5th Int. Conf. on System Reliability and Safety (ICSRS), 2021, pp. 248–257.

[15] T. Fingscheidt, H. Gottschalk, S. Houben, Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety, Springer, 2022.

[16] E. Haedecke, M. Mock, M. Akila, ScrutinAI: A Visual Analytics Approach for the Semantic Analysis of Deep Neural Network Predictions, in: Proc. EuroVis Workshop on Visual Analytics (EuroVA), 2022, pp. 73–775.

[17] C. Gladisch, C. Heinzemann, M. Herrmann, M. Woehrle, Leveraging combinatorial testing for safety-critical computer vision datasets, in: Proc. 2020 IEEE/CVF Conf. on Comp. Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 1314–1321.

[18] C. Gladisch, T. Heinz, C. Heinzemann, J. Oehlerking, A. von Vietinghoff, T. Pfitzer, Experience paper: Search-based testing in automated driving control applications, in: Proc. 2019 34th IEEE/ACM Int. Conf. on Automated Software Engineering (ASE), 2019, pp. 26–37.

[19] T. Wuellner, S. Feuerstack, A. Hahn, Clustering environmental conditions of historical accident data to efficiently generate testing sceneries for maritime systems, in: Proc. Model-Based Safety and Assessment: 6th Int. Symp., IMBSA 2019, Thessaloniki, Greece, October 16–18, 2019, Proc. 6, 2019, pp. 349–362.

[20] W.-Y. Loh, Classification and regression trees, Wiley interdisciplinary reviews: data mining and knowledge discovery 1 (2011) 14–23.

[21] G. Louppe, Understanding Random Forests: From Theory to Practive, Ph.D. thesis, University of Liège - Faculty of Applied Sciences, 2014.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An open urban driving simulator, in: Proc. 1st Annual Conf. on Robot Learning, volume 78, 2017, p. CARLA: An open urban driving simulator.

[23] I. Kurzidem, A. Saad, P. Schleiss, A Systematic Approach to Analyzing Perception Architectures in Autonomous Vehicles, in: Proc. 7th Int. Symp. on Model-Based Safety and Assessment (IMBSA), Lisbon, 2020, pp. 149–162.

[24] T. Menzel, G. Bagschik, M. Maurer, Scenarios for development, test and validation of automated vehicle, in: Proc. 2018 IEEE Intelligent Vehicles Symp. (IV), 2018, pp. 1821–1827.

[25] T. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, Machine learning 40 (2000) 139–157.