POLITECNICO DI TORINO
Repository ISTITUZIONALE

From Explainable to Reliable Artificial Intelligence

Original
From Explainable to Reliable Artificial Intelligence / Narteni, S., Ferretti, M., Orani, V., Vaccari, I., Cambiaso, Enrico., Mongelli, M.. - ELETTRONICO. - 12844:(2021), pp. 255-273. (Machine Learning and Knowledge Extraction 5th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2021 2021) [10.1007/978-3-030-84060-0_17].

Availability:
This version is available at: 11583/2966705 since: 2022-06-15T08:58:58Z

Publisher:
Springer Science and Business Media Deutschland GmbH

Published
DOI:10.1007/978-3-030-84060-0_17

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-030-84060-0_17

(Article begins on next page)

27 November 2026

From Explainable to Reliable Artificial Intelligence

Sara Narteni[1 [0000−0002−0579−647X]], Melissa Ferretti[1], Vanessa Orani[1 [0000−0003−3848−4701]], Ivan Vaccari[1 [0000−0001−7721−5737]], Enrico Cambiaso[1 [0000−0002−6932−1975]], and Maurizio Mongelli[1 [0000−0001−6201−6225]]

Consiglio Nazionale delle Ricerche - Institute of Electronics, Information Engineering and Telecommunications (CNR-IEIIT) name.surname@ieiit.cnr.it

Abstract. Artificial Intelligence systems are characterized by always less interactions with humans today, leading to autonomous decision-making processes. In this context, erroneous predictions can have severe consequences. As a solution, we design and develop a set of methods derived from eXplainable AI models. The aim is to define "safety regions" in the feature space where false negatives (e.g., in a mobility scenario, prediction of no collision, but collision in reality) tend to zero. We test and compare the proposed algorithms on two different datasets (physical fatigue and vehicle platooning) and achieve quite different conclusions in terms of results that strongly depend on the level of noise in the dataset rather than on the algorithms at hand

Keywords: Reliable AI · Logic Learning Machine · Skope Rules.

1 Introduction

Artificial Intelligence is a very wide discipline which is undergoing an unprecedented development in recent years. Algorithmic decision-making is now ubiquitous, with always less human intervention, even in critical contexts such as automotive, finance or healthcare. For this reason, there is a need for an "Algorithmic Audit" [21] facing the legal, ethical and safety issues derived from such a growth: technology experts and policy makers should cooperate in order to make AI trustworthy and responsible for users [23]. To this effort, regulation is being developed, stating the requirements that AI systems should follow to achieve such goals. Between that legislation, we must remark the European GDPR [1], introduced in 2018, which states the need of a *"right to explanation"* when dealing with automated systems. This has paved the way to the development of a subfield of AI, referred to as eXplainable AI (XAI), aiming to provide humans with understanding and trust in models outcomes. Hence, XAI models often come in the form of intelligible rules, being simpler and generally less accurate than more sophisticated models (such as those of deep learning) [35], but with the enormous advantage of being interpretable.

[1] https://gdpr.eu/tag/gdpr/

Another point of view to trustworthy AI is identifying and handling assurance under uncertainties in AI systems [11]. This means improving reliability of prediction confidence. The topic remains a significant challenge in machine learning, as learning algorithms proliferate into difficult real-world pattern recognition applications. The intrinsic statistical error introduced by any machine learning algorithm may lead to criticism by safety engineers. This is corroborated even more by the intrinsic instability of deep learning in the presence of malicious noise [39,8]. The topic has received a great interest from industry [20], in particular in the automotive [38] and avionics [9] sectors. In this context, the conformal predictions framework [3] studies methodologies to associate reliable measures of confidence with pattern recognition settings including classification, regression, and clustering.

Keeping in mind these emerging research directions, our work shows how global rule-based XAI can be used as a warranty of reliability. In particular, we give the following contributions:

- We define reliability from outside (Section 5.1) and reliability from inside (Section 5.2) methodologies, through which Logic Learning Machine characteristic value ranking becomes an instrument to achieve "safety regions" in the feature space with zero statistical error.
- We show how intelligible rules (Logic Learning Machine and Skope-Rules), when trained with zero error, can be joined and then perturbed on their most important features to obtain more complex "safety regions" (Section 5.3).
- We apply the proposed approaches on two different datasets, concerning different kinds of problems, and demonstrate how our methods may perform differently according to the data (Section 6).

2 Related Work

In the era of massive automation, a big effort must be put on developing ML/AI algorithms that should never fail when producing their outcomes: erroneous predictions may lead to severe consequences in many safety-critical fields [2]. Many different approaches have been carried out to this purpose, which will be summarized in the following subsections.

2.1 Safety Engineering-based Methods

In the context of autonomous driving, safety assessment has been studied in recent years by considering typical safety engineering approaches (safety-by-design, safe fail, safety margins) and extending them to ML paradigm [40,25], with major focus on neural networks and the most advanced Deep Learning solutions. These certification approaches include formal verification [37], transparent implementation [1], uncertainty estimation [22], error detection [16], domain generalization [43] and adversarial approaches based on data perturbation and corruption

[17,13]. Furthermore, AI certification may rely on training data quality as in [7], where authors introduced metrics such as scenario coverage for ensuring that the data used in training has possibly covered all important scenarios. Also, [15] proposed a Feature Space Partitioning Tree (FSPT) method which splits the feature space into multiple parts with different training data densities, in order to identify those where there is lack of training samples. Another work [33] adopted the same safety engineering approach to identify safety hazards related to each different phase of a typical ML pipeline and propose product-oriented (i.e. technical requirements) and process-oriented (i.e. processes to be followed) methodologies for the mitigation of such risks. In [36], authors focus on autonomous driving and review the existing machine learning safety assurance methods, categorizing them by following the system's life-cycle. Here, DNNs are massively recurrent in all the collected works, with no mention to XAI. Nowadays, most autonomous systems are based on Deep Neural Networks (DNNs), since they guarantee very accurate performance on high-dimensional data. A lot of literature exists on safety of deep models: in [13], a DNN analyzer based on abstract interpretation is introduced to enhance reliability. Safety engineering approaches are also adopted in healthcare [4] to assess Convolutional Neural Networks safety for pattern recognition using a medical device, combining the known approach of error correcting memory with the introduction of default values to use in case of uncorrectable errors. Safety of DL models is also considered in [12] by using Bayesian neural networks to quantify uncertainty of CNN models in image segmentation tasks.

Moreover, some methods integrate safety assurance into reinforcement learning (RL) framework, by making predictions to guide the agent towards safe decisions [19].

2.2 Classification with abstension

A different branch of methodologies to achieve reliability of AI consists in allowing classifiers to abstain from making predictions when they are considered uncertain according to a given loss function. Classification with abstension is achieved in [41], where a pointwise-competitive selective classification method was introduced to look for classifiers that minimize the true risk by using a selection function with the property of abstaining from predictions if the empirical risk minimizer does not agree with the true risk minimizer. Moreover, in [10] authors developed an innovative approach for classification with abstension, based on learning a predictor and the abstaining function simultaneously. Another solution is to perform a three-way decision, where an "uncertain" category is added to the task, being chosen if its cost is lower than providing a clear decision: such an approach is showing promising results either when used *a posteriori* either when embedded in the training of traditional ML [5]. However, the evaluation of such abstension-based methodologies needs to be based on a trade-off between accuracy of prediction and the rate of abstension, which cannot be too high to have useful models. In contrast, our XAI-based methods to handle uncertainty do not need such consideration.

2.3 Explainable AI-based methods

While AI systems certification is widely investigated for black-box deep learning models, it's not the same for explainable AI (XAI) models. Many XAI techniques are now available [2] with application in critical systems, e.g. in medicine [18]. In [34], the role of XAI is recognized as a way to achieve the verification of the system and the legislation compliance, but the proposed framework is based on explanations of black-boxes. Only a few works exist on the usage of XAI methods to address reliability in autonomous driving [27,29,26,28] or medicine [14]. Based on this, we investigate the role of global rule-based models and apply them to vehicle platooning and physical fatigue detection cases.

3 Logic Learning Machine

Logic Learning Machine (LLM) is an innovative global explainable supervised method; it is an efficient implementation of Switching Neural Networks [30]. LLM has the aim of building a classifier $g(x)$ described by a set of rules structured as follows: **if** <*premise*> **then** <*consequence*>. The <*premise*> is a logical product ($\wedge$) of conditions on the input features, whereas <*consequence*> corresponds to the output class. The model is built by following a three-step process:

1. *Discretization and Latticization*: each variable is transformed into a string of binary data in a proper Boolean lattice, using the inverse only-one code binarization. All the strings are then concatenated in one unique large string per each sample.
2. *Shadow Clustering*: a set of binary values, called implicants, are generated, allowing the identification of groups of points associated with a specific class.
3. *Rule Generation*: all the implicants are transformed into a set of simple conditions and eventually combined into a collection of intelligible rules.

An implicant is defined as a binary string in a Boolean lattice that uniquely determines a group of points associated with a given class. It is straightforward to derive from an implicant an intelligible rule having in its premise a logical product of threshold conditions based on the cutoffs obtained during the discretization step. In LLM all the implicants are generated via Shadow Clustering by looking at the whole training set: in this way, resulting rules can overlap and represent different relevant aspects of the underlying problem [32],[31].

3.1 Feature and Value Ranking

Being a rule-based method, it is possible to inspect LLM results through feature and value ranking.

Consider a set of m rules $\mathbf{r}_k, k = 1, \ldots, m$, each including d_k conditions $c_{l_k}, l_k = 1_k, \ldots, d_k$. Let $X_1, \ldots, X_n$ be the input variables, s.t. $X_j = x_j \in \mathcal{X} \subseteq \mathbb{R} \quad \forall j = 1, \ldots, n$. Let also $\hat{y}$ be the class assigned by the rule and y_j the real output of the $j-th$ instance.

A condition c_{l_k} involving the variable X_j, can assume one of the following forms [29]:

$$X_j > s, \quad X_j \leq t, \quad s < X_j \leq t, \tag{1}$$

being s, $t \in \mathcal{X}$.

For each rule generated by the algorithm, it is possible to define a confusion matrix associated to the rule. It is made up of four indices: $TP(\mathbf{r}_k)$ and $FP(\mathbf{r}_k)$, defined as the number of instances (x_j, y_j) that satisfy all the conditions in rule $\mathbf{r}_k$ with $\hat{y} = y_j$ and $\hat{y} \neq y_j$ respectively; $TN(\mathbf{r}_k)$ and $FN(\mathbf{r}_k)$, defined as the number of examples (x_j, y_j) which do not satisfy at least one condition in rule $\mathbf{r}_k$, with $\hat{y} \neq y_j$ and $\hat{y} = y_j$, respectively.

Consequently, the following useful metrics can be derived [6]:

$$C(\mathbf{r}_k) = \frac{TP(\mathbf{r}_k)}{TP(\mathbf{r}_k) + FN(\mathbf{r}_k)} \tag{2}$$

$$E(\mathbf{r}_k) = \frac{FP(\mathbf{r}_k)}{TN(\mathbf{r}_k) + FP(\mathbf{r}_k)} \tag{3}$$

The covering $C(\mathbf{r}_k)$ is adopted as a measure of relevance for a rule $\mathbf{r}_k$; as a matter of fact, the greater is the covering, the higher is the generality of the corresponding rule. The error $E(\mathbf{r}_k)$ is a measure of how many data are wrongly covered by the rule. Both covering and error are used to define feature ranking and the subsequent value ranking.

Feature ranking (FR) provides a way to rank the features included into the rules according to a measure of relevance. In order to obtain such measure of relevance $R(c_{l_k})$ for a condition, we consider the rule $\mathbf{r}_k$ in which condition c_{l_k} occurs, and the same rule without condition c_{l_k}, denoted as $\mathbf{r}'_k$. Since the premise part of $\mathbf{r}'_k$ is less stringent, we obtain that $E(\mathbf{r}'_k) \geq E(\mathbf{r}_k)$, thus the quantity $R(c_{l_k}) = (E(\mathbf{r}'_k) - E(\mathbf{r}_k))C(\mathbf{r}_k)$ can be used as a measure of relevance for the condition of interest c_{l_k}. Each condition c_{l_k} refers to a specific variable X_j and is verified by some values $\nu_j \in \mathcal{X}$. In this way, a measure of relevance $R_{\hat{y}}(\nu_j)$ for every value assumed by X_j is derived by the following equation 4 [29]:

$$R_{\hat{y}}(\nu_j) = 1 - \prod_k \left(1 - R\left(c_{l_k}\right)\right) \tag{4}$$

where the product is computed on the rules $\mathbf{r}_k$ that include a condition c_{l_k} verified when $X_j = \nu_j$. Since the measure of relevance $R_{\hat{y}}(\nu_j)$ takes values in $[0, 1]$, it can be interpreted as the probability that value ν_j occurs to predict $\hat{y}$. The same argument can be extended to intervals $I \subseteq \mathcal{X}$, thus giving rise to *Value Ranking (VR)*. Relevance scores are then ordered, thus giving evidence of the most sensitive interval of the feature with respect to each class.

4 Skope-Rules

Another global explainable supervised method is Skope-Rules [2], a Python machine learning module built on top of scikit-learn. Like LLM, Skope-Rules is an interpretable rule-based model consisting of a series of **if** <*premise*> **then** <*consequence*> rules; the difference between the two models lies in the way these rules are generated, selected and finally filtered. The three-step process for rules generation in Skope-Rules is as follows:

1. *Bagging estimator training*: rules generation is done from a set of decision trees and/or regressors. Each path or sub-path of a branch of a tree is transformed into a decision rule. Trees are trained to predict the output class of interest. This ensures that the splits are made in such a way as to guarantee that they are meant for the prediction task.
2. *Performance filtering*: from this set of rules an initial screening is carried out based on precision and recall thresholds.
3. *Semantic deduplication*: the last filter applied for the choice of rules is based on a criterion of similarity between terms, whereby term is meant the feature associated with the comparison operator with which it appears in the rule. The measure of similarity of two rules is determined by how many terms they have in common.

5 Reliability Assessment Methods

Considering a binary classification problem, we refer to the positive class ($y = 1$) as the unsafe one. In contrast, class $y = 0$ is referred to as the safe class. Based on this, we call "safety regions" those regions in the feature space where false negatives tend to zero. In this work, we developed three different methods to look for such regions.

5.1 Reliability from Outside

Let X be a $D \times N$ matrix of all the input vectors $x_i \in \mathbb{R}^N$, with the total number of features N and $i \in [1, D]$. Let $g(x_i) = y$ be the function describing the LLM classification. For binary classifications, we consider $g(x_i) = 1$ for the positive class, while $g(x_i) = 0$ for the other. Let D_1 be the number of instances belonging to class $y = 1$ and D_0 the number of instances in class $y = 0$, so that $D_1 + D_0 = D$.

Let N^{FR} be the number of the most significant features obtained through the feature ranking for class $y = 1$. For each feature $j \in [1, N^{FR}]$, we can use the LLM value ranking to define the most significant interval for $y = 1$ as $[s_j, t_j]$. Our method consists in expanding such intervals as follows: $[s_j - \delta_{s_j} \cdot s_j, t_j + \delta_{t_j} \cdot t_j]$.

Being $\Delta = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{N^{FR}})$ a matrix, with $\boldsymbol{\delta}_j = (\delta_{s_j}, \delta_{t_j})$, the optimal Δ is computed through the following optimization problem. Let $\mathcal{P}(\Delta)$ be the hyper-rectangle under the expanded intervals and let $\mathcal{V}(\mathcal{P}(\Delta))$ be the inherent volume.

[2] `https://github.com/scikit-learn-contrib/skope-rules`

Then, the optimization problem identifies the best fit from the outside of class $y = 1$, namely, it finds the most suitable shape, in terms of rule-based intervals, of safe points around the unsafe ones. It is as follows:

$$\Delta^* = arg \min_{\Delta: N_1 = D_1} \mathcal{V}(\mathcal{P}(\Delta)) \tag{5}$$

being N_1 the number of elements in X classified as $y = 1$ and included into $\mathcal{V}(\mathcal{P}(\Delta))$.

For instance, if we fix N^{FR}=2, the hyper-rectangle $\mathcal{P}$ becomes a rectangle $\mathcal{S}$. The optimization process let us find out the matrix $\Delta^* = (\boldsymbol{\delta}_1^*, \boldsymbol{\delta}_2^*)$. The related optimal intervals are $I_1 = (s_1 - \delta_{s_1}^* \cdot s_1, t_1 + \delta_{t_1}^* \cdot t_1), I_2 = (s_2 - \delta_{s_2}^* \cdot s_2, t_2 + \delta_{t_2}^* \cdot t_2)$, corresponding to the features $j = 1$ and $j = 2$ respectively: their logical union ($\vee$) defines a surface $\mathcal{S}$.

Then, the "safety region" is defined as the complementary bi-dimensional surface of $\mathcal{S}$, which can be written as follows:

$$\begin{aligned} \mathcal{S}_1 = ((-\infty, s_1 - \delta_{s_1}^* \cdot s_1) \vee (t_1 + \delta_{t_1}^* \cdot t_1, \infty)) \wedge \\ ((-\infty, s_2 - \delta_{s_2}^* \cdot s_2) \vee (t_2 + \delta_{t_2}^* \cdot t_2, \infty)) \end{aligned} \tag{6}$$

5.2 Reliability from Inside

An alternative way to perform the same search for "safety regions" consists in considering the N^{FR} most important features for safe ($y = 0$) class instead and reducing their most relevant intervals (again, provided by LLM value ranking) until the obtained region only contains true negative instances.

In this case, with the same notation as for the previous definition (section 5.1), the reduced intervals are: $[s_j + \delta_{s_j} \cdot s_j, t_j - \delta_{t_j} \cdot t_j]$. Being Δ defined in the same way as for equation 5 and $\mathcal{P}_0$ the hyper-rectangle under the reduced intervals, the optimal Δ is found by enlarging as much as possible the hyper-rectangle from inside the non-fatigue class, until a fatigued point is reached. It is as follows:

$$\Delta^* = arg \max_{\Delta: N_1 = 0} \mathcal{V}(\mathcal{P}_0(\Delta)) \tag{7}$$

For $N^{FR} = 2$, the "safety region" is the following rectangle $\mathcal{S}_0$:

$$\mathcal{S}_0 = (s_1 + \delta_{s_1}^* \cdot s_1, t_1 - \delta_{t_1}^* \cdot t_1) \vee (s_2 + \delta_{s_2}^* \cdot s_2, t_2 - \delta_{t_2}^* \cdot t_2) \tag{8}$$

5.3 Rules with Zero Error

As the sharp angularity of hyper-rectangles may be not fine enough to follow the potential complex shapes of the boundaries between the classes, a more refined approach would ask for more complex separators, still preserving the zero statistical error constraint and by starting from the available rule baseline.

Given a rule-based model, it can be trained so to define a set of m rules $\mathbf{r}_k,\ k = 1, \ldots, m$ denoted by $E(\mathbf{r}_k) = 0\ \forall k \in [1, m]$. Suppose that this procedure provides a set of m^0 rules $\mathbf{r}_k^0,\ k = 1, \ldots, m^0$ for the safe class ($y = 0$). Also, let $c_{l_k}^0,\ l_k^0 = (1, \ldots, d_k^0)$ be the set of d_k^0 conditions inside of each rule $\mathbf{r}_k^0$. We can join all the obtained rules $\mathbf{r}_k^0$ in logical OR operation ($\vee$), thus building a new predictor $\hat{r}$. Our goal is to assess its ability of classifying new test set data with statistical zero error (FNR=0). This implies to further tune $\hat{r}$, by tuning a subset of its conditions $c_{l_k}^0$, chosen as those containing the first N^{FR} features obtained from the rules feature ranking for class $y = 0$. In mathematical terms, for each feature $j \in [1, N^{FR}]$, we add the thresholds of the chosen conditions by applying $\boldsymbol{\delta} = (\delta_s, \delta_t)$, being δ_s and δ_t the perturbation applied to s and t thresholds, respectively, as defined in equation 1. Let $\hat{r}(\boldsymbol{\delta})$ be the resulting perturbed predictor, our goal is then to find the optimal $\boldsymbol{\delta}$ as follows:

$$\boldsymbol{\delta}^* = arg \max_{\boldsymbol{\delta}: E(\hat{r}(\boldsymbol{\delta}))=0} C(\hat{r}(\boldsymbol{\delta})) \tag{9}$$

This procedure can be applied to any rule-based model, provided that it is possible to train it with zero error.

As regards the LLM model, zero error classification (for the safe class) is readily available by the shadow clustering adopted by LLM. The clustering process is applied with the further constraint of building clusters without superposition of points of more than one class [27] (LLM 0%, in the following).

In the case of Skope-Rules (Section 4), the same zero error for safe class rules can be obtained by training the model with *precision_min* parameter fixed to 1.

6 Applications and Results

The methods described in the previous Section 5 have been applied and tested on two different classification problems: physical fatigue detection in working task simulation (Section 6.1) and collision detection in vehicle platooning (Section 6.2).

6.1 Physical Fatigue

The data used in this test phase belong to an open-source dataset [3]. Data were collected through wearable sensors, i.e. Inertial Movement Units (IMUs), from 15 participants who were asked to perform a simulation of an industrial task for 180 minutes and provide a fatigue level every 10 minutes using RPE [42]. According to such scale, RPE$\geq$13 corresponds to a fatigued state (class $y = 1$), otherwise to non-fatigued (class $y = 0$). From sensors raw data, a list of features is derived (see Table 2 in [24]). We removed heart-rate related features as well as gender, since it is not numerical, and standardized data by applying z-score transformation.

[3] `https://github.com/zahrame/FatigueManagement.github.io/tree/master/Data`

We then trained LLM model with standard 5% maximum error allowed for rules on a 67% training set. We evaluated it on a 33% test set using common metrics, namely an accuracy of 82%, sensitivity of 71%, specificity of 95% and F1-score of 0.81.

Reliability from Outside In order to test this method, we considered the first two most important intervals for fatigued class that we got from LLM value ranking: *back rotation position in sagittal plane* > 0.03 and *wrist jerk coefficient of variation* > 0.03. We applied the optimization algorithm (Eq. 5) on such intervals and obtained $\delta^*_{s_1} = -13, \delta^*_{s_2} = 28$. For such values, we got FNR=0 and TNR=0.20. Therefore, the "safety region", which we call "non-fatigue region" in this context, can be expressed as follows (for brevity, let f_1 and f_2 be the two above mentioned features):

$$\mathcal{S}_1 = ((f_1 \in (-\infty, 0.42)) \wedge (f_2 \in (-\infty, -0.81))$$

The resulting region was then validated in order to take into account that the involved feature values should vary in a limited range, so to reflect real human movement capabilities and correspond to proper execution of the task. In general, we cannot assume that a subject who stays still will not ever get fatigued, but the nature of the task in which the subject is involved should provide indications on the ranges of parameters assessing the required movements. Since the dataset documentation does not drive in this direction and the inherent literature lacks of standard ranges, we chose to consider maximum and minimum values for the features based on two age groups (age$\leq$40 and age$>$40). This helps to highlight the further stratification readily available from the sensitivity analysis.

Doing so, we were able to redefine two "non-fatigue regions" by limiting the previous one according to the ranges we found; such new regions are expressed as follows:

$$\mathcal{S}_1 = ((f_1 \in (-2.52, 0.42)) \wedge (f_2 \in (-1.78, -0.81)) \text{ for } age \leq 40 \ y.o$$

$$\mathcal{S}_1 = ((f_1 \in (-1.86, 0.42)) \wedge (f_2 \in (-2.0, -0.81)) \text{ for } age > 40 \ y.o$$

In Figure 1 a visual representation of the obtained regions is provided.

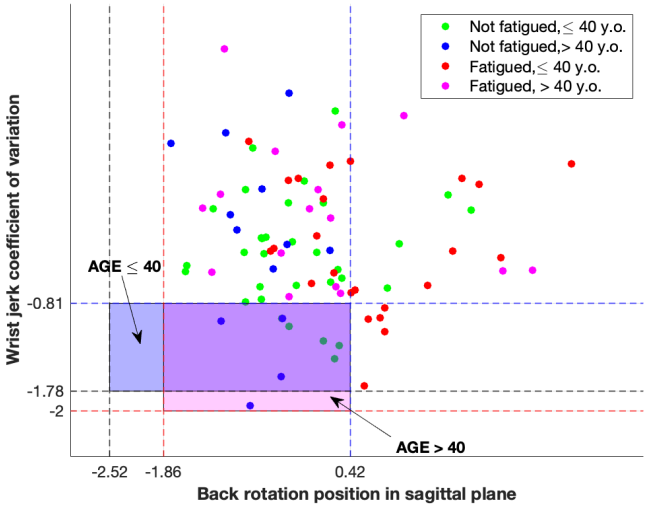


Fig. 1. Scatter plot of the first two features (back rotation position in sagittal plane and wrist jerk coefficient of variation) with representations of the "non-fatigue region" (FNR=0) individuated for age ≤ 40 group (pink) and age > 40 (violet).

Reliability from Inside We considered the problem of identifying non-fatigue regions starting from the non-fatigued class too, thus adopting the reliability from inside approach. The value ranking shown *back rotation position in sagittal plane* $\leq$ 0.03 and *chest acceleration mean* $>$ -0.47 as the two most relevant intervals for predicting non-fatigued class. On such conditions, we applied the optimization problem (Eq. 7), which led us to individuate $\delta^*_{t_1} = 57$, $\delta^*_{s_2} = 8.78$. For these values, we got FNR=0 and TNR=0.06. The "non-fatigued region" $\mathcal{S}_0$ is then found (with f_1 and f_2 being *back rotation position in sagittal plane* and *chest acceleration mean* respectively):

$$\mathcal{S}_0 = (f_1 \in (-\infty, -1.68) \vee f2 \in (3.65, \infty))$$

Just as for the outside approach, we limited such region in function of the two group ages (up to and over 40 years old). This procedure redefines $\mathcal{S}_0$ for the two age groups as follows (see Fig. 2 for the graphical representation):

$$\mathcal{S}_0 = (f_1 \in (-2.52, -1.68) \vee f2 \in (3.65, 3.99)) \; for \; age \leq 40 \; y.o.$$
$$\mathcal{S}_0 = (f_1 \in (-1.86, -1.68) \vee f2 \in (3.65, 3.99)) \; for \; age > 40 \; y.o.$$

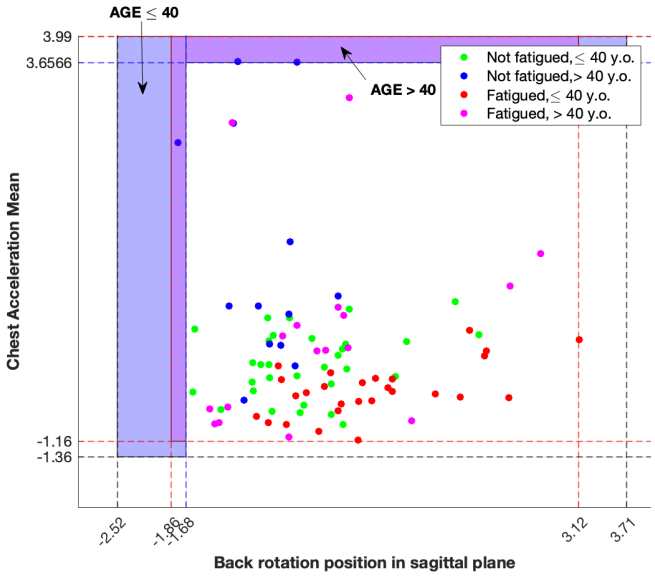


Fig. 2. Scatter plot of the first two features (back rotation position in sagittal plane, Chest Acceleration Mean) from value ranking of non-fatigued class, with representations of the "non-fatigue regions" (FNR=0) based on the age group (violet for age $\leq$ 40, pink otherwise)

Zero Error LLM Both the previous approaches have the limitation of individuating optimal solutions to the identification of "non-fatigue regions" characterized by relatively low values of TNR, i.e. number of instances included in such surfaces.

In order to assess if such values could be increased, we trained the LLM 0% and built a new predictor by joining the first four highest coverage rules in logical OR (see below).

if (0.51 < HipACCMean $\leq$ 1.98 **and** ChestACCcoefficientofvariation $\leq$ 1.11 **and** -1.73 < averagestepdistance $\leq$ 0.81 **and** backrotationpositioninsagplane $\leq$ 0.52) $\vee$
(WristjerkMean > 0.55 **and** -1.35 < Back rotation position in sag plane $\leq$ 0.04) $\vee$
(-1.73 < averagestepdistance $\leq$ -0.22 **and** backrotationpositioninsagplane $\leq$ -0.25 **and** -0.44 < numberofsteps $\leq$ 3.75 **and** -1.73 < Wristjerkcoefficientofvariation $\leq$ 0.55) $\vee$
(ChestxpostureMean > -0.033 **and** HipzpostureMean > 0.43 **and** WristACCMean > -0.83 **and** -0.88 < backrotationpositioninsagplane $\leq$ 0.29)
then non-fatigued

By evaluating the joining before any perturbation, we got FNR=0.06 and TNR=0.75. To further decrease the FNR, we conducted the optimization process

described in equation 9 by tuning the thresholds for the first $N^{FR} = 2$ features from non-fatigued feature ranking, namely *HipACCMean* and *WristjerkMean*. We obtained $\delta^*_{s_1}$ =1.848 and $\delta^*_{t_2}$ =0.027 for such features respectively: these thresholds perturbations brought FNR=0.02, with TNR=0.42.

Skope-Rules To ensure that we obtained rules with zero errors on the non-fatigue classification task, we trained several models with a *precision_min* = 1, where *precision_min* is the parameter that defines the minimum precision of a rule to be selected in the *performance filtering*. Trained models differ in *n_estimators* and *max_depth_duplication*, where *n_estimators* is the number of base estimators to use for prediction and *max_depth_duplication* is the the maximum depth of the decision tree for *semantic deduplication* (Section 4). For each model thus obtained, we calculated precision and recall by varying the number of rules applied (from 2 up to the maximum number of rules generated by the model) and then chose the one that maximised precision and recall. This led us to use a model trained with the following parameters:

1. *n_estimators* = 200
2. *precision_min* = 1
3. *max_depth_duplication* = 5

We then chose the first 3 rules generated by this model which correspond to the following logical OR ($\vee$):

if (backrotationpositioninsagplane $\leq$ 0.08 **and** HipjerkMean $>$ -1.03 **and** HipACCcoefficientofvariation $\leq$ 0.75 **and** HipypostureMean $\leq$ 1.12 **and** HipzpostureMean $>$ -1.78) $\vee$
(backrotationpositioninsagplane $\leq$ 0.17 **and** Wristjerkcoefficientofvariation $\leq$ 0.05 **and** HipACCMean $>$ -0.47) $\vee$
(backrotationpositioninsagplane $\leq$ 0.22 **and** Wristjerkcoefficientofvariation $\leq$ 0.06 **and** HipACCMean $>$ -0.10 **and** ChestjerkMean $>$ -1.36) **then** **non-fatigued**

This new predictor, before applying any perturbation, leads to FNR=0.11 and TNR=0.69. As in the previous case, let's see what happens in terms of FN by perturbing two features. The features we are going to perturb are *backrotation-positioninsagplane* and *Wristjerkcoefficientofvariation* and they are respectively the first and second most present features in the rules derived from the performance filtering (Section 4). To carry out the perturbation we used the procedure as described in Section 5.3, applying the method of Eq. 9 and perturbing only the most restrictive thresholds when the same features appeared in more than one rule. This leads us to the following suboptimal solution, with an FNR=0.07 and TNR=0.67, corresponding to $\delta_{t1} = 1.717$ for *backrotationpositioninsagplane* and $\delta_{t2} = 15.845$ for *Wristjerkcoefficientofvariation*.

6.2 Vehicle Platooning

Vehicle platooning is one of the most important challenges in autonomous driving, dealing with a trade-off between performance and safety. In our analysis

we considered a scenario of cooperative adaptive cruise control (CACC) as described in [27], where the platoon is in a steady state of speed and reciprocal inter-vehicular distance when a braking force is applied by the leader of the platoon. For the application of our safety assessment methods we used simulation data generated by Plexe simulator [4]. For each of the 4744 generated samples, 5 features were computed within the following ranges: the number of vehicles, $N \in [3, 8]$ the braking force $F_0 \in [-8, -1] \times 10^3$ N the Packet Error Rate $PER \in [0.2, 0.5]$ the initial distance between vehicles $d(0) \in [4, 9]$ m (supposed equal for all of them); the initial speed $v(0) \in [10, 90]$km/h. The system registers a collision when distance between two vehicles is lower than 2 m.

Applying the default LLM with maximum error of 5% on a 30% test set, we obtained 85,9% of accuracy, 75.4% sensitivity, 86.8% specificity and 0.46 F1-score. We then performed the safety analysis to find out regions were collisions are avoided with no error.

Reliability from Outside From the value ranking for the collision class ($y = 1$), we obtained $PER >$0.43 and $F_0 \leq -7.50 \times 10^3$N as the first two most important intervals. We then applied the optimization approach as in Eq. 5 and found $\delta^*_{s_1} = -0.034$, $\delta^*_{t_2} = -0.416$, which correspond to reach FNR=0 with TNR=0.34. Thus, according to the definition in Eq. 6, the safety region we obtain is the following:

$$\mathcal{S}_1 = ((PER \in (0.2, 0.4154)) \wedge (F_0 \in (-4.37, -1) \times 10^3)$$

A visual representation of such region is in Figure 3. Also, we performed a search for safety regions by considering three features, including the third most important interval from value ranking too, i.e. $N > 6$. We got $\delta^*_{s_1} = -0.184, \delta^*_{t_2} = -0.166$ and $\delta^*_{s_3} = -0.1$ with FNR=0 and TNR=0.19. In this case, the safety region is tridimensional, corresponding to the following volume (Fig. 4):

$$\mathcal{V}_1 = ((PER \in (0.2, 0.3509)) \wedge (F_0 \in (-6.255, -1) \times 10^3) \wedge (N \in (3, 5.4)$$

Reliability from Inside Following the optimization approach in Eq. 7, we first chose the first two intervals from the value ranking of the safe class ($y = 0$): $PER \leq 0.33$ and $F_0 > -3.50 \times 10^3$N. Then, we computed the optimal threshold perturbations $\delta^*_{t_1} = 0.356, \delta^*_{s_2} = 0.686$, for which we got FNR=0 with TNR=0.13. The safety region is then individuated by the following surface (Fig. 5):

$$\mathcal{S}_0 = (PER \in (0.2, 0.2125) \vee F_0 \in (-1.1001, -1) \times 10^3)$$

Zero Error LLM By lowering the LLM maximum error allowed to 0% we were able to look for more complex safety regions. After training the LLM model with 0% error, we joined the first 4 rules for safe class with the highest coverage. This corresponded to the following logical OR ($\vee$):

[4] `https://github.com/mopamopa/Platooning`

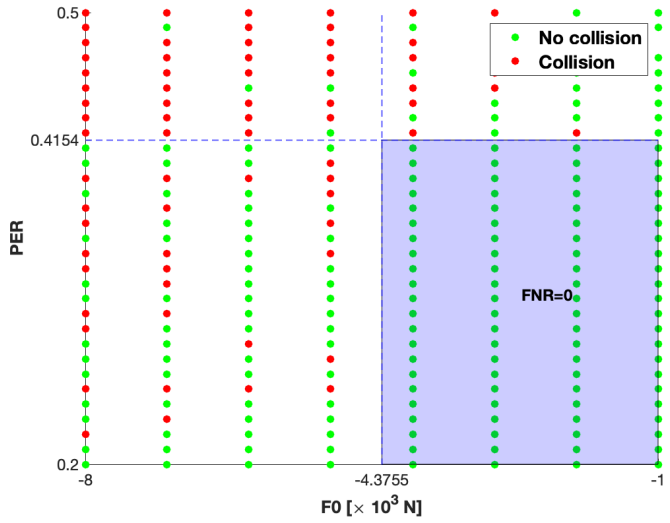


Fig. 3. Scatter plot of the first two features (PER and F0) with representations of the safety region

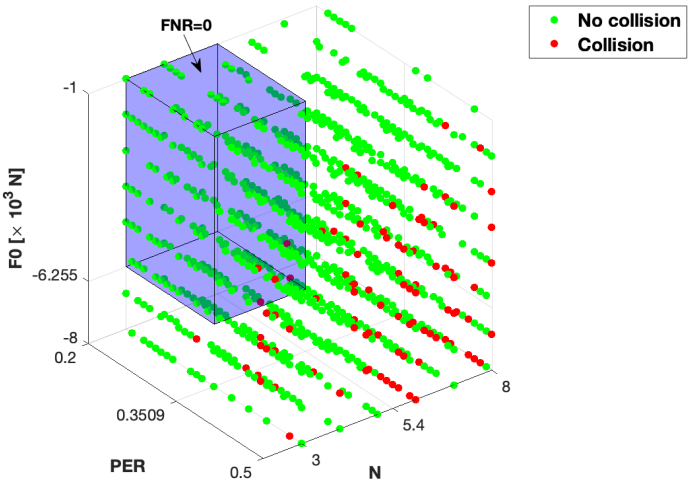


Fig. 4. 3D scatter plot of the first three features (PER,F0,N): the safety region is represented by the volume (in violet)

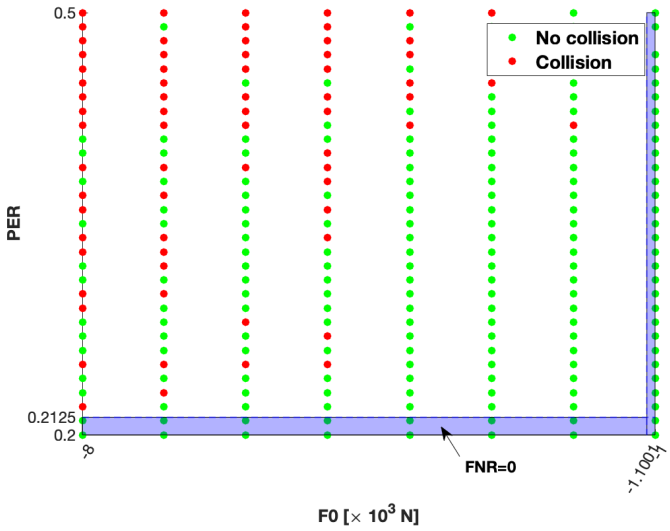


Fig. 5. Scatter plot of the first two features (PER and F0) for safe class with representations of the safety region

$$\textbf{if } (N \leq 5 \textbf{ and } v(0) \leq 54.50) \vee$$
$$(PER \leq 0.295 \textbf{ and } N \leq 7 \textbf{ and } v(0) \leq 86.50) \vee$$
$$(v(0) \leq 28.50 \textbf{ and } PER \leq 0.445) \vee$$
$$(v(0) \leq 28.50 \textbf{ and } N \leq 6 \textbf{ and } d(0) \leq 7.86) \textbf{ then safe}$$

This new predictor, before applying any perturbation, leads to FNR=0.05 and TNR=0.55. We then exploited the feature ranking to individuate which features we should tune in order to lower FNR as much as possible. The two most influent features resulted to be $v(0)$ and PER in this case. Then, by applying the method in Eq. 9 we perturbed such features: in this case, we were able to achieve only a suboptimal solution, with FNR=0.02 and TNR=0.45, corresponding to $\delta_{t1} = 0.000877$ for $v(0)$ and $\delta_{t2} = 0.277$ for PER. Where the same feature was present in more than one joined rule, we perturbed only the most stringent threshold.

Skope-Rules As explained above for the Physical Fatigue case, also for Platooning, we trained different models by varying the parameters *n_estimators* and *max_depth_duplication*. Again, we chose to set *precision_min* = 1 to obtain rules with zero errors on the non-collision classification task. Again, for each model thus obtained, we calculated precision and recall by varying the number of rules applied (from 2 up to the maximum number of rules generated by the model) and then chose the one that maximised precision and recall. This led us to use a model trained with the following parameters:

1. *n_estimators* = 75
2. *precision_min* = 1

3. *max_depth_duplication* = 2

We then chose the first 4 rules generated by this model which correspond to the following logical OR ($\vee$):

$$\textbf{if } (PER \leq 0.41 \textbf{ and } v(0) \leq 45.5) \vee$$
$$(N \leq 7.5 \textbf{ and} F0 > -7.5 \textbf{ and} PER \leq 0.32) \vee$$
$$(N \leq 5.5 \textbf{ and } v(0) \leq 54.5) \vee$$
$$(F(0) > -4.5 \textbf{ and } PER \leq 0.41 \textbf{ and } v(0) > 64.5) \textbf{ then safe}$$

This new predictor, before applying any perturbation, leads to FNR=0.04 and TNR=0.57. To compare the results obtained previously for the Zero Error LLM, we decided again to perturb two features in the same way as described above (applying the method of Eq. 9 and perturbing only the most restrictive thresholds). In this case, the first and second most present features in the rules derived from the performance filtering (Section 4) are *v0* and *PER*, the same obtained from the LLM ranking. This leads us to the following suboptimal solution, with an FNR=0.02 and TNR=0.52, corresponding to $\delta_{t1} = -0.649$ for *v(0)* and $\delta_{t2} = -0.172$ for *PER*.

6.3 Discussion

From a comparison between the obtained results on the two datasets, we can notice that inferring reliability from the available rules is highly dependent on the structure of the data under analysis. The inside-outside (Sections 5.2, 5.1) methods show flexibility in looking at the feature space, alternating good results (outside in platooning in two dimensions), surprising results (outside in platooning in three dimensions is outperformed by the same in two dimensions) and bad results (inside in platooning in two dimensions). The outside approach finds larger (higher TNR) safety regions than the inside one both in fatigue and platooning. Inside-outside may be even joined together when the feature ranking agrees on the most important features for the available classes. As this happens in the platooning case, we may consider the safety regions involving PER and F_0 (Figures 3 and 5), and, by visual analysis of the overlap of such regions (see Fig. 6), we could join them to find a larger and more complex (in terms of rules) safety region.

On the other hand, due to the similarity of the adopted rules optimization approach (Section 5.3), we can compare the results of LLM 0% and Skope-Rules. Since we were dealing with more complex profiles than rectangles, results have shown an increase of TNR for both the models on the two datasets. However, in the physical fatigue test case, the LLM 0% starts by a much lower FNR (0.06) than Skope (0.11) before perturbation, reaching a sub-optimal solution after tuning; in contrast, Skope achieves a suboptimal solution too, with a FNR (0.07) that is surprisingly higher than the corresponding value of LLM 0% before optimization (0.06). As regards the vehicle platooning problem, results are more consistent in the two algorithms, showing the same FNR and a higher TNR with Skope.

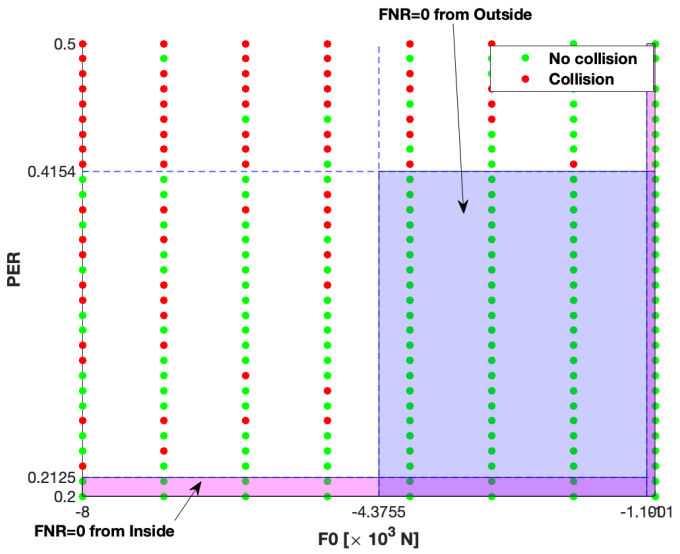


Fig. 6. Scatter plot of the two most important features in vehicle platooning LLM classification (PER and F_0), with representation of the safety regions found with Inside (pink area) and Outside (blue area) methods: the overlap of such regions defines a new safety region, where TNR reaches higher values

7 Conclusions and Future Works

In this work, we have studied how XAI models can represent a solution towards safety assurance in predictive analytics. We first focused on a global rule-based model, the LLM, and demonstrated how its characteric value ranking property can be exploited for the design of "safety regions" in the features space with zero statistical error. This was achieved by developing our innovative "reliability from outside" and "reliability from inside" methodologies. Then, we used a third method to optimize more complex rule profiles and applied it to LLM 0% and Skope-Rules.

Data and code are available at the following Github repository: `https://github.com/saranrt95/safety-from-valueranking`.

By testing and comparing our proposed methodologies on problem instances of different nature (physical fatigue and vehicle platooning), we have also shown how their performance varies between the datasets.

Future works may extend the testing through cross-validation in the presence of a large amount of data, including the adoption of data augmentation techniques and the experimentation on benchmark datasets. The characterization of the placement of the points deserves further study to understand the optimal covering of the safety regions. The translation of deep learning logic into rules with further design of safety envelope is another topic we are going to pursue in the near future.

References

1. Adebayo, J., et al.: Sanity checks for saliency maps. arXiv preprint arXiv:1810.03292 (2018)
2. Arrieta, A.B., et al.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion **58**, 82–115 (2020)
3. Balasubramanian, V.N., Ho, S., Vovk, V.: Conformal Prediction for Reliable Machine Learning. Morgan Kaufmann Elsevier, 225 Wyman Street, Waltham, MA 02451, USA, 1 edn. (2014)
4. Becker, U.: Increasing safety of neural networks in medical devices. In: International Conference on Computer Safety, Reliability, and Security. pp. 127–136. Springer (2019)
5. Campagner, A., et al.: Three-way decision for handling uncertainty in machine learning: A narrative review. In: Bello, R., Miao, D., Falcon, R., Nakata, M., Rosete, A., Ciucci, D. (eds.) Rough Sets. pp. 137–152. Springer International Publishing (2020)
6. Cangelosi, D., et al.: Logic learning machine creates explicit and stable rules stratifying neuroblastoma patients. BMC bioinformatics **14**(7), 1–20 (2013)
7. Cheng, C.H., et al.: Towards dependability metrics for neural networks (2018)
8. Clavière, A., Asselin, E., Garion, C., Pagetti, C.: Safety verification of neural network controlled systems. arXiv preprint arXiv:2011.05174 (2020)
9. Cluzeau, J., Henriquel, X., Rebender, G., Soudain, G., van Dijk, L., Gronskiy, A., Haber, D., Perret-Gentil, C., Polak, R.: Concepts of design assurance for neural networks codann. Standard, European Union Aviation Safety Angency, Daedalean, AG (Mar 2020), also available as `https://www.easa.europa.eu/sites/default/files/dfu/EASA-DDLN-Concepts-of-Design-Assurance-for-Neural-Networks-CoDANN.pdf`
10. Cortes, C., et al.: Boosting with abstention. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 29. Curran Associates, Inc. (2016), `https://proceedings.neurips.cc/paper/2016/file/7634ea65a4e6d9041cfd3f7de18e334a-Paper.pdf`
11. Czarnecki, K., Salay, R.: Towards a framework to manage perceptual uncertainty for safe automated driving. In: International Conference on Computer Safety, Reliability, and Security. pp. 439–445. Springer (2018)
12. Eaton-Rosen, Z., et al.: Towards safe deep learning: accurately quantifying biomarker uncertainty in neural network predictions. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 691–699. Springer (2018)
13. Gehr, T., et al.: Ai2: Safety and robustness certification of neural networks with abstract interpretation. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 3–18. IEEE (2018)
14. Gordon, L., et al.: Explainable artificial intelligence for safe intraoperative decision support. JAMA surgery **154**(11), 1064–1065 (2019)
15. Gu, X., Easwaran, A.: Towards safe machine learning for cps: infer uncertainty from training data (2019)
16. Guo, C., et al.: On calibration of modern neural networks. In: International Conference on Machine Learning. pp. 1321–1330. PMLR (2017)

17. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations (2019)
18. Holzinger, A., et al.: What do we need to build explainable ai systems for the medical domain? (2017)
19. Isele, D., et al.: Safe reinforcement learning on autonomous vehicles. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–6. IEEE (2018)
20. ISO/IEC: Standardization in the area of artificial intelligence. Standard, ISO/IEC, Washington, DC 20036, USA (Creation date 2017), `https://www.iso.org/committee/6794475.html`
21. Koshiyama, A., et al.: Towards algorithm auditing: A survey on managing legal, ethical and technological risks of ai, ml and associated algorithms. SSRN Electronic Journal (2021)
22. Lakshminarayanan, B., et al.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6405–6416 (2016)
23. Madhavan, R., et al.: Toward trustworthy and responsible artificial intelligence policy development. IEEE Intelligent Systems **35**(5), 103–108 (2020)
24. Maman, Z.S., et al.: A data analytic framework for physical fatigue management using wearable sensors. Expert Systems with Applications **155**, 113405 (2020)
25. Mohseni, S., et al.: Practical solutions for machine learning safety in autonomous vehicles. arXiv preprint arXiv:1912.09630 (2019)
26. Mongelli, M., Muselli, M., Ferrari, E.: Achieving zero collision probability in vehicle platooning under cyber attacks via machine learning. In: 2019 4th International Conference on System Reliability and Safety (ICSRS). pp. 41–45. IEEE (2019)
27. Mongelli, M., Ferrari, E., Muselli, M., Fermi, A.: Performance validation of vehicle platooning through intelligible analytics. IET Cyber-Physical Systems: Theory & Applications **4**(2), 120–127 (2019)
28. Mongelli, M., Muselli, M., Scorzoni, A., Ferrari, E.: Accellerating prism validation of vehicle platooning through machine learning. In: 2019 4th International Conference on System Reliability and Safety (ICSRS). pp. 452–456. IEEE (2019)
29. Mongelli, M., Orani, V.: Stability certification of dynamical systems: Lyapunov logic learning machine. In: International Conference on Applied Soft computing and Communication Networks (ACN20) (2020)
30. Muselli, M.: Switching neural networks: A new connectionist model for classification (2005)
31. Parodi, S., et al.: Differential diagnosis of pleural mesothelioma using logic learning machine. BMC bioinformatics **16**(9), 1–10 (2015)
32. Parodi, S., et al.: Logic learning machine and standard supervised methods for hodgkin's lymphoma prognosis using gene expression data and clinical variables. Health informatics journal **24**(1), 54–65 (2018)
33. Pereira, A., Thomas, C.: Challenges of machine learning applied to safety-critical cyber-physical systems. Machine Learning and Knowledge Extraction **2**(4), 579–602 (2020)
34. Samek, W., et al.: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services **1**, 1–10 (2017)
35. Saranti, A., et al.: Property-based testing for parameter learning of probabilistic graphical models. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 499–515. Springer (2020)

36. Schwalbe, G., Schels, M.: A survey on methods for the safety assurance of machine learning based systems. In: 10th European Congress on Embedded Real Time Software and Systems (ERTS 2020) (2020)
37. Seshia, S.A., et al.: Formal specification for deep neural networks. In: International Symposium on Automated Technology for Verification and Analysis. pp. 20–34. Springer (2018)
38. for Standardization, I.O.: Road vehicles safety of the intended functionality pd iso pas 21448:2019. Standard, International Organization for Standardization, Geneva, CH (Mar 2019)
39. Sun, Y., et al.: Structural test coverage criteria for deep neural networks. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). pp. 1–23. ACM New York, NY, USA (2019)
40. Varshney, K.R.: Engineering safety in machine learning. In: 2016 Information Theory and Applications Workshop (ITA). pp. 1–5. IEEE (2016)
41. Wiener, Y., El-Yaniv, R.: Agnostic pointwise-competitive selective classification. J. Artif. Int. Res. **52**(1), 179–201 (2015)
42. Williams, N.: The borg rating of perceived exertion (rpe) scale. Occupational Medicine **67**(5), 404–405 (2017)
43. Zhang, X., et al.: Dada: Deep adversarial data augmentation for extremely low data regime classification. IEEE Transactions on Circuits and Systems for Video Technology pp. 2807–2811 (2019)