Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio

doi: 10.3850/978-981-18-8071-1_ESREL2023-P697-cd

Failure On Demand Analysis in the Case of Score Based Binary Classifiers: Method and Application

Alexander Günther
Software Engineering, Rheinland-Pfälzische Technical University Kaiserslautern-Landau, Germany.
E-mail: alexander.guenther@cs.uni-kl.de

Peter Liggesmeyer
Franhofer Institue for Experimental Software Engineerung, Kaiserslautern, Germany.
Software Engineering, Rheinland-Pfälzische Technical University Kaiserslautern-Landau, Germany.

Safety assessment and verification have become more complex in the past years. Especially the incorporation of machine learning components, and their black box nature, are proposing new difficulties to overcome. Therefore new techniques are needed to judge the safety of machine learning components and further integrate those into existing safety analysis methods. In this contribution we will provide a new method for safety analysis of a score based binary classifier. The presented technique can output a single-value estimation for the failure on demand. Latter one can then be used inside a system safety analysis, as done for physical engineering systems. In particular we will briefly mention a general approach for score based binary classifiers, as already applied for general systems. Furthermore we will contribute a more refined method in the case of a normal distributed score. The main idea is to incorporate confidential bounds on the parameters to obtain a function that serves as an estimated upper bound for the failure on demand. Further analysis of the retrieved function will then provide a mathematically based single-value estimation for the reliability. In the end of this work we will demonstrate this technique at the example of breast cancer detection and evaluate the performance in this scenario.

Keywords: Safety Analysis, Failure, Binary Classifier, Normal Distribution, Score, Confident Bounds.

1. Introduction

In the last decade machine learning was one of the most evolving fields in computer science. Stunning results have been achieved in a wide range of tasks, across a manifold of fields. For example genome classification in biology, Remita et al. (2017). Or Regression inside the area of earth observations, Huber et al. (2022). Even in a sparse domain, the classical pipelines of feature engineering, followed by a machine learning component have been applied successfully Jin et al. (2020). Not only super superhuman performance is possible, also domain experts can be outperformed by machine learning, as shown by De Fauw et al. (2018).

With all these positive examples of machine learning there also arises the wish to use the latter ones in safety-critical systems. Due to their complexity most advanced methods are black box models and therefore predictions and outputs are hard to explain or verify. Furthermore, the remaining uncertainty of systems is difficult to measure, especially in open-context applications. Nevertheless, we still want to have a quantified measure of the remaining aleatoric and epistemic uncertainty. Or different speaking, reliable values for the remaining reducible and irreducible uncertainty, Hüllermeier and Waegeman (2019). In this paper, we will therefore compute the probability of failure or more precisely an estimated upper bound for the probability of failure. This is highly important to judge a system verification, in particular under the classical risk acceptance approaches like As Low As Reasonable Possible, Globalement Au Moins Aussi Bon, and Minimal Endogenous Mortality. To obtain these values is possible, even in the case of a black box model, and is presented by Lucas et al. (2008). But they also have shown that the number of necessary test cases can be infeasible high in practice, for example in the case of complete black box components. Thus it is important to individually tailor the safety assessment to the properties of each system. Then tighter and more important, reliable, failure rates can possibly be achieved. Within this paper, we contribute to this pool of techniques by analyzing score-based binary classifiers. In particular, the case of a normal distributed score is the major focus of this work. On top of that we are going to demonstrate the techniques in case of breast cancer recognition.

2. Method

First, we will clarify the notation and our definition of score score-based binary classifier. We denote the data points with $x \in \mathbb{R}^n$ and the score function with $s : \mathbb{R}^n \to \mathbb{R}$. In this context a score-based binary classifier f is then given by

$$f(x) = H \circ s(x).$$

The function H is the Heaviside step function. Alternatively a general step function over an arbitrary interval $(-\infty, c]$ could be used, which is equivalent to this case. Simply because every classifier of that kind can be converted, by a shift of the score function, to the one above. In this setting, a linear support vector machine is given by $s(x) = \langle w, x \rangle + b$, where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are obtained at training.

For a binary classifier, there are only two possible failures that could occur, namely classifying a data point X^*, which we will represent as random variable (RV), from class 1 to class 0 and vice versa. If we denote our probability space with $(\Omega, \mathcal{A}, P)$, we can also directly state the probability of that occurrence as

$$P(f(X^*) = 0) = P(s(X^*) < 0). \tag{1}$$

Further, we assume that our given data are from the same underlying probability distribution. This distribution could be different for each class and also possibly unknown.

2.1. *Measure Concentration Estimates*

As described by Lucas et al. (2008), one can use concentration of measure inequalities to obtain an upper estimate for the probability of failure. The critical aspect of this approach is the number of necessary test data that need to be examined in order to retrieve a reliable statement. Another important factor can be the image set of their performance measure, in particular, we need to know the admissible set. On top of that, Lucas et al. (2008) has shown that as more information about the system is known, as closer the upper estimate possibly is to the real values. Consequently, we impose additional information on the system.

2.2. *Normal distributed Score*

Next, we consider the situation when $s(X^{(1)}) \sim \mathcal{N}(\mu_1, \sigma_1)$, so when the score is normal distributed for one class, here we choose 1. Also, class 0 could be normal distributed, which we will treat in subsection 2.2.4 since it is a little different. Mathematically we view $s(X)$ as independent and identically distributed RVs, and we will also use capital letters for the RVs. This might arguably be two strong assumptions but the normal distribution plays such an important role in theory and practice, the latter one is apposite to consider. Further, this assumption is for instance fulfilled if the input data are multivariate normal and we are considering a plain Support Vector Machine, a basic machine learning scenario. In contrast to other approaches, like Braband and Schäbe (2020), we only put assumptions on the score and not on the input. Nevertheless, a probabilistic view of the data is necessary to encounter aleatoric uncertainty as well. Next, we use the estimators

$$\bar{\mu}_1 = \frac{1}{N_1} \sum_{j=1}^{N_1} s\left(X^{(j)}\right), \tag{2}$$

$$\bar{\sigma}_1 = \sqrt{\frac{1}{N_1 - 1} \sum_{j=1}^{N_1} \left(s\left(X^{(j)}\right) - \bar{\mu}_1\right)^2}. \tag{3}$$

As a result equation (1) can easily be computed by

$$P(s(X^*) < 0) \approx \Phi\left(-\frac{\bar{\mu}_1}{\bar{\sigma}_1}\right). \tag{4}$$

Clearly, the estimate depends on the event $\omega \in \Omega$, but here and in the following we will omit this in our notation to not get overly lengthy. As we still have not considered the uncertainty of the estimators themselves, we are going to include confidence bounds for those as well.

2.2.1. *Confidence Bounds on Estimators*

To include confidence bounds one can refer to any statistical book, like Georgii (2015) (Satz 9.17), to obtain that $\bar{\mu}_1$ and $\bar{\sigma}_1$ are independent. Additionally, for confidence γ, η in $(0,1)$, probabilistic bounds are given by

$$P\left(\bar{\mu}_1 - t_{N_1-1}(1-\gamma) \cdot \frac{\bar{\sigma}_1}{\sqrt{N_1}} \leq \mu_1\right) = 1-\gamma \tag{5}$$

$$P\left(\sigma_1^2 \leq \frac{N_1-1}{\chi_{N-1}(1-\eta)}\right) = 1-\eta, \tag{6}$$

where χ_{N-1} is the inverse of the cumulative distribution function from a χ^2-distribution with $N_1 - 1$ degrees of freedom and t_{N-1} is the inverse of $F_{t_{N_1-1}}(1-\cdot)$ where $F_{t_{N_1-1}}$ is the cumulative distribution function for the students t-distribution with $N_1 - 1$ degrees of freedom. To save space we will denote the event in equation (5) with M and the event described in (6) with S.

Now we only need two technical assumptions to state the resulting probability. First that $\bar{\mu}_1 > 0$, which is a mathematical formulation that the classifier is to some extent sufficient to solve the task. The second one is

$$-\bar{\mu}_1 + t_{N_1-1}(\gamma)\frac{\bar{\sigma}_1}{\sqrt{N_1}} < 0. \tag{7}$$

For fixed γ this will be fulfilled for large N_1 due to lemma 2.1, or in any case where $\gamma \in \left(\frac{1}{2}, 1\right)$. It now holds for any $\omega \in \Omega$ that

$$\begin{aligned} P(s(X^*) < 0) &\leq P(s(X^*) \leq 0) \cdot \mathbb{1}_{M \cap S}(\omega) \\ &\quad + \underbrace{P(s(X^*) \leq 0)}_{=:\,P_0 \leq 1} \mathbb{1}_{M^C \cup S^C}(\omega). \end{aligned} \tag{8}$$

Next, we are going to take the expected value. As $\mathbb{E}[\mathbb{1}_E] = P(E)$, for a measurable event E, we get

$$\begin{aligned} &P(s(X^*) < 0) \\ &\leq \mathbb{E}\left[\Phi\left(\underbrace{\frac{-\bar{\mu}_1 + \frac{t_{N_1-1}(\gamma)\bar{\sigma}_1}{\sqrt{N_1}}}{\sqrt{\frac{N_1-1}{\chi_{N_1-1}(\eta)}}\bar{\sigma}_1}}_{=:\,\nu(\gamma,\eta)}\right)\right](1-\gamma)(1-\eta) \\ &+ \quad ((1-\gamma)\cdot\eta + \gamma\cdot(1-\eta) + \gamma\cdot\eta). \end{aligned} \tag{9}$$

To make use of this upper bound we need to estimate the expected value in (9). We will do so by using $\Phi(\nu(\gamma,\eta))(\omega)$, so the observed estimations. As this is only an approximation, the hardness of this bound can mathematically not be guaranteed anymore. But as we will see later in 2.2.7 the bound seems a good estimate.

We denote with

$$\begin{aligned} g^{(N)}(\gamma,\eta) = \Phi&\left(\frac{-\bar{\mu}_1(\omega) + \frac{t_{N_1-1}(\gamma)\bar{\sigma}_1(\omega)}{\sqrt{N_1}}}{\sqrt{\frac{N_1-1}{\chi_{N_1-1}(\eta)}}\bar{\sigma}_1(\omega)}\right) \\ &\cdot (1-\gamma)(1-\eta) \\ &+ ((1-\gamma)\cdot\eta + \gamma\cdot(1-\eta) + \gamma\cdot\eta), \end{aligned} \tag{10}$$

the estimated upper bound function, also depending on ω. In cases where the dependency of N is less important, we will drop the N in notation. As long as condition (7) is fulfilled, g serves as an estimated upper bound for valid parameters $(\gamma,\eta) \in (0,1)^2$.

Lemma 2.1. *The sequence* $\{|t_{N-k}(\gamma)|\}_N$ *is bounded for every fixed* $\gamma \in (0,1)$. *Hence*

$$\lim_{N\to\infty} \frac{t_{N-k}(\gamma)}{\sqrt{N}} = 0, \quad \forall k \in \mathbb{N}. \tag{11}$$

2.2.2. *Analysis of the Upper Bound Function*

Next, we have a short analytical look at the function g. We will only state the developed theorems without proof. Simply to not exceed the scope of this paper and to not get lost in the technicalities.

Theoretically interesting is that the function g provides an estimated upper bound for most parameters, while the actual failure rate is independent of those. Therefore we can simply take the minimum of g, if it exists, as our estimated failure on demand. This can be obtained by any method of choice. In our simulation and application example we used the heuristic given in algorithm 1, together with the gradient descent method in each variable. This approach was chosen because g is differentiable and the gradients can be explicitly computed, as given in lemma 2.2. Additionally theorems 2.1, 2.2 show that the function g, viewed as a univariate function, for a fixed ω, hold unique minima.

Theorem 2.1. *The function* $g_{1,\gamma_0} : (0,1) \to \mathbb{R}, \eta \mapsto g(\gamma_0, \eta)$ *with* $\gamma_0 \in (0,1)$ *has a minimum for* N *large enough. Furthermore the restriction* $g_{1,\gamma_0}\big|_{(0,c)}$ *is convex, for every* $c < \frac{1}{2}$.

Theorem 2.2. *The function* $g_{2,\eta_0} : (0,1) \to \mathbb{R}, \gamma \mapsto g(\gamma, \eta_0)$ *with* $\eta_0 \in (0,1)$ *has a global minimum for* N *large enough. Furthermore the restriction* $g_{2,\eta_0}\Big|_{\left(1-F_{t_{N-1}\left(\sqrt{N}\frac{\bar{\mu}}{\bar{\sigma}}\right)}, \frac{1}{2}\right)}$ *is convex.*

Lemma 2.2. *The function g is twice differentiable and the derivatives are given as follows*

$$\begin{aligned}\frac{\partial g}{\partial \gamma}(\gamma_0, \eta_0) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\nu(\gamma_0, \eta_0)^2}{2}\right) \\ &\cdot \left(\frac{\partial \nu}{\partial \gamma}(\gamma_0, \eta_0)\right) \cdot (1-\gamma_0)(1-\eta_0) \\ &- \Phi(\nu(\gamma_0, \eta_0))(1-\eta_0) + (1-\eta_0), \quad (12)\end{aligned}$$

$$\begin{aligned}\frac{\partial g}{\partial \eta}(\gamma_0, \eta_0) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\nu(\gamma_0, \eta_0)^2}{2}\right) \\ &\cdot \left(\frac{\partial \nu}{\partial \eta}(\gamma_0, \eta_0)\right) \cdot (1-\gamma_0)(1-\eta_0) \\ &- \Phi(\nu(\gamma_0, \gamma_0))(1-\gamma_0) + (1-\gamma_0), \quad (13)\end{aligned}$$

where the derivative of ν *is given as*

$$\begin{aligned}\frac{\partial \nu}{\partial \gamma}(\gamma_0, \eta_0) &= -\sqrt{\pi} \cdot \sqrt{\frac{\chi_{N-1}(\eta_0)}{N}} \cdot \frac{\Gamma\left(\frac{N-1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \\ &\cdot \left(1 + \frac{t_{N-1}(\gamma_0)^2}{N-1}\right)^{\frac{N}{2}}, \quad (14)\end{aligned}$$

$$\begin{aligned}\frac{\partial \nu}{\partial \eta}(\gamma_0, \eta_0) &= \frac{2^{\frac{N-3}{2}} \Gamma\left(\frac{N-1}{2}\right)}{\sqrt{N-1}} \\ &\cdot \left(-\frac{\bar{\mu}}{\bar{\sigma}} + \frac{t_{N-1}(\gamma_0)}{\sqrt{N}}\right) \cdot \chi_{N-1}^{1-\frac{N}{2}}(\eta_0) \\ &\cdot \exp\left(\frac{\chi_{N-1}(\eta_0)}{2}\right). \quad (15)\end{aligned}$$

2.2.3. *Numerical Issues*

The term $\frac{\Gamma\left(\frac{N_1-1}{2}\right)}{\Gamma\left(\frac{N_1}{2}\right)}$ will cause some numerical problems for big values since both tend to infinity.

Algorithm 1 Heuristic to find minima

Require: $\bar{\mu} > 0, \bar{\sigma} > 0, N > 0, \varepsilon > 0$
Ensure: Return value is a valid upper bound
$\gamma \leftarrow \frac{1}{4}$ ▷ Initialize γ
$\eta \leftarrow \frac{1}{4}$ ▷ Initialize η
$m \leftarrow g(\gamma, \eta)$ ▷ Initialize minimum function value
change $\leftarrow \infty$ ▷ Variable to store current change
repeat
  $\gamma \leftarrow \operatorname{argmin}_{\tilde{\gamma} \in (0,1)} g_{2,\eta}(\tilde{\gamma})$ ▷ For instance with gradient descent or Newton's method
  $\eta \leftarrow \operatorname{argmin}_{\tilde{\eta} \in (0,1)} g_{1,\gamma}(\tilde{\eta})$ ▷ For instance with gradient descent or Newton's method
  change $\leftarrow |m - g(\gamma, \eta)|$ ▷ Update current change
  $m \leftarrow g(\gamma, \eta)$ ▷ Update current minimum
until change $< \varepsilon$
if $-\bar{\mu} + \frac{t_{N-1}(\bar{\gamma})\bar{\sigma}}{\sqrt{N}} < 0$ **then**
  return m
else
  return 1
end if

This problem can be overcome with an approximation. We define the sequences

$$l_N := \begin{cases} \frac{1}{\left\lfloor \frac{N-1}{2} \right\rfloor} \sqrt{\left\lfloor \frac{N-1}{2} \right\rfloor - \frac{1}{4}}, & \text{if } N \text{ is even,} \\ \frac{1}{\left\lfloor \frac{N-1}{2} \right\rfloor} \sqrt{\left\lfloor \frac{N-1}{2} \right\rfloor + \frac{1}{4}}, & \text{if } N \text{ is odd,} \end{cases} \quad (16)$$

$$u_N := \begin{cases} \frac{1}{\left\lfloor \frac{N-1}{2} \right\rfloor} \sqrt{\left\lfloor \frac{N-1}{2} \right\rfloor}, & \text{if } N \text{ is even,} \\ \frac{1}{\left\lfloor \frac{N-1}{2} \right\rfloor} \sqrt{\left\lfloor \frac{N-1}{2} \right\rfloor + \frac{1}{2}}, & \text{if } N \text{ is odd.} \end{cases} \quad (17)$$

Then with Gautschi's-inequality and equation (1.2) form Alzer (1993) one can prove that l_N is a lower bound and u_N is an upper bound for the quotient. As the difference of these bounds tends to zero we retrieve the following lemma 2.3.

Lemma 2.3. *The sequences* l_N *and* u_N *are ap-*

proximating $\frac{\Gamma\left(\frac{N-1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)}$, *so formally it holds that*

$$\lim_{N\to\infty}\left|l_N-\frac{\Gamma\left(\frac{N-1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)}\right|=0 \tag{18}$$

$$\lim_{N\to\infty}\left|u_N-\frac{\Gamma\left(\frac{N-1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)}\right|=0. \tag{19}$$

We recommend using l_N as the approximation since it seems to be more tight than the mean of l_N and u_N.

2.2.4. *Other Side Error*

Interestingly the same mathematical approach for the vice versa error, namely to classify a data point of class 0 to class 1, is more challenging. The estimates will not be applicable in the same way. Nevertheless, there is an easy possibility to also use the approach above, by “mirroring” the classifier. Therefore we look at the classifier with score function $s'(x)=-s(x)$, which outputs the same results, only with switched classes. Now the analysis from above can be applied and returns the desired value. At first look, one may think that points with a score exactly zero can cause problems. This is, at least theoretically, not the case. For exact mirroring, we need to slightly modify the Heaviside step function to also map 0 to value 0. Then the condition in (1) and (8) changes to a smaller equal. Nevertheless, since we have a continuous probability distribution, equation (9) stays unchanged.

2.2.5. *The influence of N*

In this paragraph, we will briefly mention the effect of increasing the amount of test data. The intuition that increasing N also reduces the remaining uncertainty is reflected in the upper bound function. Corollary 2.1 shows that mathematically the upper bound function is point-wise decreasing, and therefore the upper bound for the failure probability must decrease. Only numerical issues, like for instance an inaccurate determination of the minimum, can cause this effect to be false.

Corollary 2.1. *For fixed* $\gamma_0,\eta_0 \in \left(0,\frac{1}{2}\right)$, *there exists* $N_0 \in N$ *such that the sequence* $\left(g^{(N)}(\gamma_0,\eta_0)\right)_{N\geq N_0}$ *is monotone decreasing.*

2.2.6. *Comparison to Estimator*

We want to mention, that the use of these confidence bounds on the estimators will always reflect in a higher bound than to use the estimations alone.

Corollary 2.2. *For all* $\omega\in\Omega$ *and* $\gamma,\eta\in(0,1)^2$ *it holds that*

$$g^{(N)}(\gamma,\eta)\geq\Phi\left(-\frac{\bar{\mu}_1}{\bar{\sigma}_1}\right). \tag{20}$$

2.2.7. *Validity of Estimated Upper Bound*

To verify that the estimate holds reliable results, we will simulate the underlying mathematical distribution and check whenever the bound is valid. In particular we simulate the normal score $s(X^*)$ for the parameters combinations $\mu,\sigma=\{0.01+j0.01 \mid j=0,\ldots,1.99\}$ and $n=\{100,102,\ldots,300\}$. The computations are carried out in Matlab R2023a, while the function “normrnd” was used for simulation, and the simulation was repeated if the mean estimation was small or equal to zero. Further, we repeated the experiment for 100 times. We computed the minimum of g with the heuristic 1 and gradient descent. The individual learning rate is divided by 2 whenever the last step goes outside the variable range, so below 0 or above 1, or has lead to a deterioration. In this case, also the step was reverted. This continues until the change is smaller 10^{-6}. Finally condition (7) was checked, in case of non-fulfillment the upper bound 1 was used.

As a result, in $6.2255\cdot10^{-4}$ percent of the cases, the bound was not valid and bigger then $\Phi\left(-\frac{\mu}{\sigma}\right)$.

3. Simulation

The goal of this simulation is to verify the applicability, even under noisy data. To do so we used Matlab R2023a to simulate two-variate normal distributed data.

3.1. *Outline and Parameters*

For each class, we used 1000 data points for training, as seen in figure 1, and 200 points for testing. The mean and covariance matrices C_i are given as

$$\mu_1=\begin{pmatrix}-1.0\\-3.0\end{pmatrix},\quad C_1=\begin{pmatrix}1.0 & 0.0\\0.0 & 2.0\end{pmatrix}, \tag{21}$$

$$\mu_2 = \begin{pmatrix} 1.0 \\ 1.2 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 3.0 & 1.0 \\ 1.0 & 1.0 \end{pmatrix}. \qquad (22)$$

Additionally, we added noise in the form of white Gaussian noise samples of power -15, with the internal Matlab function “wgn”. Afterwards, we used the provided function “fitcsvm” to train a linear Support Vector Machine.

3.2. *Evaluation*

The resulting score should by the underlying mathematical theory, be uni-variate normal distributed. Therefore we performed a Jarque-Bera-Test (JB-Test) and Anderson-Darling-Test (AD-Test) with 5% Significance, which confirmed the normality. A visual validation in the form of a histogram is printed in figure 1, which also shows the influence of the noise. Afterward we used the method described in section 2, to compute the potential minimum of our upper bound function. In particular, we used heuristic 1 together with a gradient descent method, where the individual learning rates were divided by 1.75, respectively 2 for the other side error, whenever the last step goes outside the variable range. The failure rates are visualized in table 1. The total rate of falsely classified points is 0.025. As we see in the difference between the False Classification rate (FCR) and the Probabilistic Evaluation upper bound with Estimator (PEE), the introduction of this probabilistic framework does not necessarily increased the error rate. This underlines the necessity of bounds on the estimators. The use of the confident bounds on the estimators, instead of the estimator usage only, seems to have a strong effect and is reflected in the Upper Bound with Confidence (UBC) rate. That possibly originates in the low number of test data, and therefore perfectly mirrors the uncertainty contained. In the opposite scenario, where sufficient test data is given, we expect the value to decrease, as already theoretically confirmed in 2.1. For the other side error, FCR$=0.02$, PEE$=0.042835$ and UBC$=0.097715$, show similar behaviour.

Table 1. The failure rate and upper bounds for falsely classifying a data point of class 1 to class 0.

Failure Verification Type	Rate
False Classification Rate (FCR)	0.03
Probabilistic Evaluation with Estimators (PEE)	0.02682
Upper Bound with Confidence (UBC)	0.07262

To confirm the influence of N, the same simulation has been carried out with 300 test data points from each class. Afterwards the failure rates for misclassification of a data-point from class 1 has been computed on the first 150 points only. Iteratively 50 have been added and the failure rates have been recomputed. The results are shown in table 2 and confirm the expected behaviour. Nevertheless, the randomness of these simulations is particular visible in the FCR.

As mentioned in the beginning the goal was to validate the applicability. For the numerical backup of the theoretically proven properties, even for bigger N, another simulation study needs to be done. Additionally the influence of noise needs further investigations.

Table 2. The failure rates FCR, PEE and UBC for falsely classifying a data point of class 1 to class 0 under increasing amount of test data.

Number of Test Points (N)	FCR	PEE	UBC
150	0.0600	0.0517	0.1225
200	0.0650	0.0452	0.1029
250	0.0600	0.0384	0.0862
300	0.0533	0.0359	0.0790

4. Application: Breast Cancer Prediction

In this application, we are solving the task of breast cancer prediction. Therefore we use the Wisconsin Breast Cancer Database (January 8, 1991), which is published at Wolberg (1995) and initially used in Mangasarian and Wolberg (1990). It holds 699 data samples, of which 483 are benign and 241 are malignant. Of these samples, 16 had

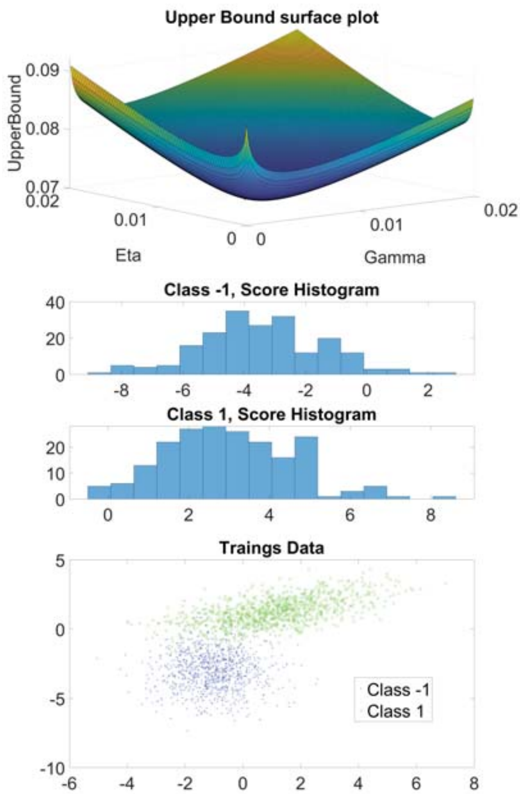

Fig. 1. At the top is a surface plot, together with the computed minimum as red dot, for the resulting upper bound function, given in equation 9, for class 1 of the simulation data. In the middle are the histograms of the score displayed for each class. At the bottom is a scatter plot of the training data.

missing values therefore only 683 had been used. We encountered all 9 features, which are in the range of $[0, 10]$. Detailed information is also given at Wolberg (1995) and not provided here. We randomly split each class into half for training and used the other half for testing and computations. As classifier, a linear support vector machine has been used, with an overall false classification rate of 0.032164.

Our next step is the analysis of the failure on demand and an estimated upper bound for this occurrence. The pipeline, presented in the next subsection, can be adapted to any other application task, which fulfills the assumptions.

4.1. *Evaluation*

Next similar analytical steps as in section 3 have been performed. First, the distribution of the score has been tested for normality with JB-Test at 5% Significance. The score for class 0, which corresponds to benign, is not normal distributed, but the score for class 1, which corresponds to malignant, is normal distributed. See also figure 2 for visual validation. The failure bounds have been computed with the heuristic 1, together with a gradient descent method, where the individual learning rates were divided by 1.75, whenever the last step goes outside the variable range. If we look at the resulting bounds in table 3, we see the same qualitative behavior as in our simulations and expectations from the underlying model. The introduction of random input increases the theoretical failure. Further, the introduction of confidential bounds on the parameters is increasing the upper failure rate. Shortly to mention that a non-normal score leads to a breakdown of the proposed method, as seen in table 3, for the benign class. Even if the UBC rate might seem correct it is only a consequence of aleatoric uncertainty and not reliable.

Table 3. The failure rate and upper bounds for falsely classifying a data point.

Failure Verification	false benign classification	false malignant classification
FCR	0.045045	0.0083333
PEE	0.029812	0.015642
UBC	0.075203	0.063974

5. Discussion and Future Work

In this work, we presented a new technique to compute an estimated upper bound for the error on demand, namely in the case of normal distributed score for score-based binary classifier. We evaluated the correctness of our method in a simulation which confirmed the theoretical computations. Additionally, we provided an application in breast cancer classification, which backed up

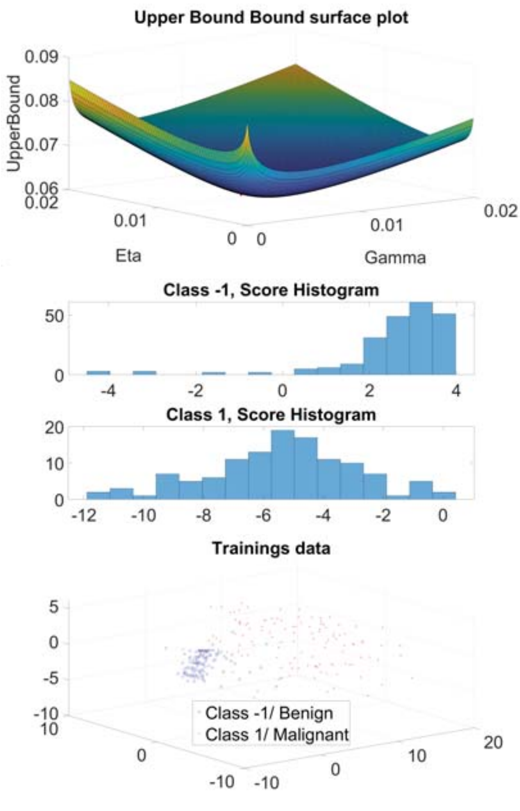

Fig. 2. At the top is a surface plot, together with the computed minimum as red dot, for the resulting upper bound function, given in equation 9, for false malignant classification. In the middle are the histograms of the score, displayed for each class. At the bottom is a scatter plot of the training data. For visualization a dimensional reduction to 3 dimensional has been performed with the Matlab function “pca”.

the simulations and has shown that the property of normal distribution is needed and can not be relaxed. It still is an open question on how tight the upper bound really is and if it is close enough to the actual probability of failure. Furthermore, it still is unknown how many real applications have a score of approximately normal distribution and how “close” to a normal distribution the score has to be, in order to provide reliable results. As we have seen the technique breaks down if this assumption is violated, so one may ask: how close is close enough? In particular, for practitioners, it might be interesting at which significance level the normality test has to be performed so that the technique performs well. The latter question can possibly be answered in further simulations.

Acknowledgement

For everyone who supported this work by actively hinting at literature, commenting, or in any other form: Thank you.

References

Alzer, H. (1993). Some gamma function inequalities. *Mathematics of Computation 60*(201), 337–346.

Braband, J. and H. Schäbe (2020, dec). On safety assessment of artificial intelligence. *Dependability 20*(4), 25–34.

De Fauw, J., J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O’Donoghue, D. Visentin, G. van den Driessche, B. Lakshminarayanan, C. Meyer, F. Mackinder, S. Bouton, K. Ayoub, R. Chopra, D. King, A. Karthikesalingam, C. O. Hughes, R. Raine, J. Hughes, D. A. Sim, C. Egan, A. Tufail, H. Montgomery, D. Hassabis, G. Rees, T. Back, P. T. Khaw, M. Suleyman, J. Cornebise, P. A. Keane, and O. Ronneberger (2018, Sep). Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature Medicine 24*(9), 1342–1350.

Georgii, H.-O. (2015). *Stochastik*. Berlin, München, Boston: De Gruyter.

Huber, F., A. Yushchenko, B. Stratmann, and V. Steinhage (2022). Extreme gradient boosting for yield estimation compared with deep learning approaches.

Hüllermeier, E. and W. Waegeman (2019). Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction. *CoRR abs/1910.09457*.

Jin, X.-S., J. Li, and X. Du (2020, nov). Image classification of chicken embryo based on matched filter and skeleton curvature feature. *Journal of Physics: Conference Series 1651*(1), 012196.

Lucas, L., H. Owhadi, and M. Ortiz (2008). Rigorous verification, validation, uncertainty quantification and certification through concentration-of-measure inequalities. *Computer Methods in Applied Mechanics and Engineering 197*(51), 4591–4609.

Mangasarian, O. L. and W. H. Wolberg (1990). Cancer diagnosis via linear programming. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Remita, M. A., A. Halioui, A. A. Malick Diouara, B. Daigle, G. Kiani, and A. B. Diallo (2017, Apr). A machine learning approach for viral genome classification. *BMC Bioinformatics 18*(1), 208.

Wolberg, William, S. W. . M. O. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. DOI: `10.24432/C5DW2B`.