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
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_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 reliable value 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 upper bound for the failure on demand. Further analysis of the retrieved function will then provide a mathematically based single value 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 inside 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 there area of earth observations, Huber et al. (2022). Even in sparse domain the classical pipelines of feature engineering, followed by a machine learning component has been applied successfully, Jin et al. (2020). Not only super human performance is possible, also domain experts can be outperformed by machine learning, as shown by De Fauw et al. (2018).

With all this positive examples of machine learning there also raises the wish to use 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 are difficult to measure, specially 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 precise an 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 (ALARP), Globalement Au Moins Aussi Bon (GAMAB) and Minimal Endogenous Mortality (MEM). To obtain these values is possible, even in the case of a black box models, 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 tight and more important, reliable, failure rates can possibly be achieved. Within this paper we contribute to this pool of techniques by analysis score based binary classifier. In particular the case of a normal distributed score is the major focus inside this work. On top of that we are going to demonstrate the techniques in the case of breast cancer recognition.

2. Method

First we will clarify the notation and our definition of 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^* from class 1 to class 0 and vice versa. We can also directly state the probability of that occurrence as

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

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

2.1. *Measure Concentration Estimates*

As described by Lucas et al. (2008), one can use concentration of measure inequalities to obtain a upper estimates for the probability of failure. The critical aspect with this approach is the number of necessary test data that need to be examined in order to retrieve a reliable statement. Another important factor is the image set of the input data variables. In particular we need to know the space of this input features or a way to obtain the verification diameter of the system. On top of that, Lucas et al. (2008) have shown that as more information about the system is known, as closer the upper estimate possibly are to the real values. Consequently we impose an 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 random variables, and we will also use capital letter for the random variable. This might arguably be two strong assumptions but since 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 on the data is necessary to encounter for aleatoric uncertainty as well. Next we use the standard 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

$$P(H(s(x^*)) \leq 0) = \Phi\left(-\frac{\bar{\mu}_1}{\bar{\sigma}_1}\right). \tag{4}$$

Here we still have not considered the uncertainty of the estimators itself, so we are going to include confidence bound for those as well.

2.2.1. *Confidence Bounds on Estimators*

To include confidence bounds one can refer 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 extend 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 $\gamma \in \left(\frac{1}{2},1\right)$, due to the previous assumption. Finally we obtain the upper bound function, as

$$\begin{aligned} &P(s(x^*)\leq 0) \qquad (8)\\ &= P(s(x^*)\leq 0 \mid M\wedge S)\cdot P(M\wedge S)\\ &+ \underbrace{P(s(x^*)\leq 0 \mid \neg M\vee\neg S)}_{=:\,P_0\leq 1} P(\neg M\vee\neg S) \qquad (9)\\ &\leq \Phi\left(\frac{-\bar{\mu}_1 + \frac{t_{N_1-1}(\gamma)\bar{\sigma}_1}{\sqrt{N_1}}}{\underbrace{\sqrt{\frac{N_1-1}{\chi_{N_1-1}(\eta)}}\bar{\sigma}_1}_{=:\,\nu(\gamma,\eta)}}\right)(1-\gamma)(1-\eta)\\ &+ ((1-\gamma)\cdot\eta + \gamma\cdot(1-\eta) + \gamma\cdot\eta). \qquad (10)\end{aligned}$$

We denote the function in equation 10 with g, or $g^{(N)}$ if we want to stress the dependence on N. As long as condition (7) is fulfilled, g serves as an upper for any 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.

Theoretical interesting is that the function g provides an upper bound for almost all parameters, while the actual failure rate is independent of those. Therefore we can simply take the minimum of g, if it exits, as our 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. We repeatable computed the minimum in each variable via gradient decent method, until convergence. This approach was chosen because g is differentiable and the gradients can be explicit computed, as given in lemma 2.2. Additionally theorems 2.1, 2.2 show that the function g, viewed as uni variate function, hold unique minima.

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

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

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

where the derivative of ν is given as

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

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

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. This problem can be overcome with an approximation. We denote 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 tend to zero we retrieve the following lemma 2.3.

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
  $\gamma \leftarrow \underset{\tilde{\gamma}\in(0,1)}{\operatorname{argmin}} g_{2,\eta}(\tilde{\gamma})$ ▷ For instance with gradient descent or Newton's method
  $\eta \leftarrow \underset{\tilde{\eta}\in(0,1)}{\operatorname{argmin}} g_{1,\gamma}(\tilde{\eta})$ ▷ 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 $-\bar{\mu}+\frac{t_{N-1}(\gamma)\bar{\sigma}}{\sqrt{N}} < 0$ **then**
  return m
else
  return 1
end if

Lemma 2.3. *The sequences l_N and u_N are approximating $\frac{\Gamma\left(\frac{N-1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)}$, so formally it holds that*

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

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

We recommend to use l_N as the approximation since it seems to be more tight then 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. On first look one may think that the points with score exactly zero can cause problems. This is, at least theoretically, not the case. For exact mirroring we can slightly modify the Heaviside step function to also map 0 to value 1. Then the condition in (8) and (9) changes to strictly smaller. Nevertheless, since we have a continuous probability distribution, equation (10) stays unchanged.

2.2.5. *The influence of N*

In the end of this section we will shortly mention the effect of increasing amount of test data. The intuition that increasing N also reduces the remaining uncertainty does reflect in the upper bound function. Corollary 2.1 shows that mathematically the upper bound function is point-wise decreasing, and therefore also 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.*

3. Simulation

The goal of this simulation is to verify the applicability, even under noisy data. To do so we used Matlab (R2022a) 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

$$\mu_1 = \begin{pmatrix} -1.0 \\ -3.0 \end{pmatrix}, C_1 = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 2.0 \end{pmatrix}, \quad (20)$$

$$\mu_2 = \begin{pmatrix} 1.0 \\ 1.2 \end{pmatrix}, C_2 = \begin{pmatrix} 3.0 & 1.0 \\ 1.0 & 1.0 \end{pmatrix}. \quad (21)$$

Additionally we added noise in 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 form of a histogram is printed in figure 1, which also shows the influence of the noise. Afterwards 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 decent method. The only difference is that we computed step 7 with the old γ values, in order to update the shared learning rate for the next iteration. All failure rates are visualized in table 1. The total rate of falsely classified points is 0.025. As we see in the difference of False Classification rate (FC) and the Probabilistic Evaluation upper bound with Estimator (PEE), the introduction of this probabilistic framework increases the error rate. This is only natural as additional to epistemic uncertainty, the PEE captures also aleatoric uncertainty. Further the use of confident bound on the estimators, instead of the estimator usage only, seems to have a strong effect and reflects in the Upper Bound with Confidence (UPC) rate. That possibly originates in the low number of test data, and therefore perfectly mirrors the uncertainty contained. In the opposite scenario, where

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 (FC)	0.03
Probabilistic Evaluation with Estimators (PEE)	0.041835
Upper Bound with Confidence (UPC)	0.098434

sufficient test data is given, we expect the value to decrease, as already theoretically confirmed in 2.1.

In table 2 small variations in N are displayed and confirm the theoretical statement. Only the value for $N = 300$ doesn't seem to fit but might also be strongly influenced by the problematic quotient. Additionally in this case, PEE=0.053325 and FC=0.053333 has increased as well. This could be the consequence of noise, since we can directly see in the histograms in figure 1 that the noise has a strong influence. However this example shows that even unexpected increases in the failure rate, will reflect in the UPC.

As mentioned in the beginning the goal was to validate the applicability and therefore noise can not be neglected. For the numerically back up of the theoretical proven properties, another simulation study needs to be done. We did investigate this relation only for small values in this concrete example because our implementation uses the exact computation of the quotient described in subsection 2.2.3. As the focus these simulation was different, the avoidance of numerical problems was primarily important.

Table 2. The failure rate and upper bounds for falsely classifying a data point of class 1 to class 0 und different amount of test data.

Number of Test Points (N)	Upper Bound with Confidence (UPC)
150	0.10737
200	0.098434
250	0.094702
300	0.10583

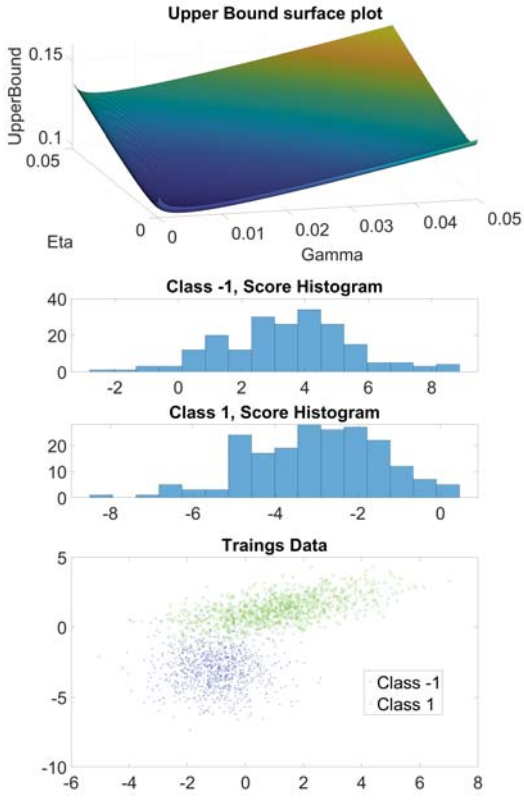

Fig. 1. At the top is a surface plot for the resulting upper bound function, given in equation 10, for class 1 of the simulation data. In the middle are the histograms of the score displayed for each class. At the bottom is a scatter plot of the training data.

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 publish 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 missing values therefore only 683 had been used. We encountered all 9 features, which are in the range of $[0, 10]$. Detailed information are 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 a overall false classification rate of 0.032164.

Our next step is the analysis of the failure on demand and an 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 the same 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 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. If we look at the resulting bounds in table 3, we see the same qualitive behaviour as in our simulations and expectations from the underlying model. The introduction of random input increases the theoretical failure, as it also encounters for aleatoric uncertainty. Further the introduction of confidential bounds on the parameters is increasing the upper failure rate. Shortly to mention is 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 UPC 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
FC	0.045045	0.0083333
PEE	0.015642	0.029812
UPC	0.052513	0.096433

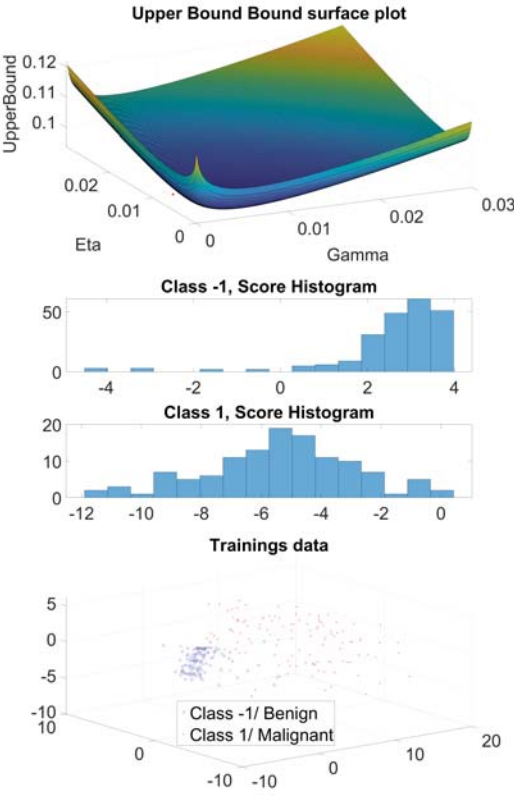

Fig. 2. At the top is a surface plot for the resulting upper bound function, given in equation 10, for false malignant classification. In the middle are the histograms of the score, displayed for each class. At the bottom is a scatter plot of the training data. To make a visualization possible a dimensional reduction to 3 dimensional has been performed with the Matlab function “pca”.

5. Discussion and Future Work

In this work we presented a new technique to compute an 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 an simulation which confirmed the theoretical computations. Additionally we provided an application in breast cancer classification, which backed up the simulations and have 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 have to be performed, so that the technique performs well. Latter question can possibly be answered in further simulations.

Acknowledgement

For everyone who supported this work by actively hinting 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.