Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p26306-cd

Single-valued Risk Bound for Normal Classifiers

Alexander Günther

Chair Software Engineering: Dependability, University of Kaiserslautern-Landau, Germany.
E-mail: alexander.guenther@rptu.de

Peter Liggesmeyer

Franhofer Institue for Experimental Software Engineerung (IESE), Germany.
Chair Software Engineering: Dependability, University of Kaiserslautern-Landau, Germany.
E-mail: peter.liggesmeyer@iese.fraunhofer.de

Machine Learning has entered almost every aspect of our lives, particularly in software systems. Their performance in several tasks has reached human-level and is therefore interesting to use in safety-critical systems. However, the black-box character of most advanced models hinders the use of many classical assessment techniques. This makes it more difficult to acquire the crisp failure rates needed in techniques like fault trees, reliability block diagrams or partially FMCEA. In this contribution, we aim to provide a bridge from testing to an estimation of an upper bound. This estimated delimiter of the model's risk can then be used in established techniques and tools to obtain early estimations. That is particularly useful in early design stages to foster a feedback loop about planned system designs. Our method is suited for any classifier with a normal distributed score. The estimations are established with the use of confidence bounds on the parameters. More specifically, we are going to propose a function, which has the confidence levels as variables, that serves as an upper bound estimation. In a follow-up step, the minimum of this function can be used as an upper bound estimation, which is given by a single number. On top of that, the minimizing arguments can be seen as an optimal choice of the confidence levels. In a simulation, we are going to prove the validity of this attempt. Additionally, we are going to demonstrate its application with the example of breast cancer detection.

Keywords: machine learning, risk assessment, confidence selection, upper bound estimation, reliability, normal distribution.

1. Introduction

Machine Learning solutions have been getting increasingly popular in the last decade and have reached human-level performanceEsteva et al. (2017). That makes them interesting for use in safety-critical systems as well. However, the common software testing approaches, like MCDC Chilenski and Miller (1994) to name one, are not directly applicable. Their neural network counterparts Sun et al. (2019) and other approaches like Pei et al. (2019) still remain heuristics in terms of formal safety guarantees. This leaves the question about which quantitative value to use in subsequent risk assessments. Using the measured test performance would have a chance of underestimating and creating an unsafe system. Adding a safety margin or using a confidence bound Bousquet et al. (2004) opens the question about the size of the margin or which confidence level. Inside this paper, we are going to answer the question in the case of normal score classifiers.

1.1. *Related Work*

This work was inspired by Braband and Schäbe (2020) and is built upon their idea to use the confidence bound of the estimators. However, they perform a budget calculation, and we are going to compute an upper bound estimation that can be viewed as an optimal or automatic selection of the confidence levels.

That certification by testing is theoretically possible was proven by Lucas et al. (2008), inside the framework of Quantification of Margin and Uncertainties Pilch et al. (2011). However, in our setup, the margin is unknown and needs to be determined for future risk assessment. Additionally, the results inside this contribution do not aim at

certification, but rather as supporting information in early design stages.

In the case of general score and dependent data, this has already been attempted in Günther et al. (2025). The authors have built an upper bound function, depending on the margin to the measured performance and retrieve the optimum as the minimizer. This will be similar in our case, except that we are going to use confidence levels of the parameter estimators.

Another alternative to this approach, if we include the risk assessment process, can be the use of fuzzy numbers Dijkman et al. (1983) and fuzzy sets Zadeh (1965). See Kabir and Papadopoulos (2018) for an overview in this regard. This, however, leaves the field of probability theory towards possibility theory Zadeh (1999).

1.2. *Contribution*

This paper provides a method, presented in Section 2, to obtain an upper bound estimation on the multi-class classification risk, given that the score is normal distributed. Simultaneously, this procedure can be used to obtain an automatic confidence level selection for the parameter estimators. Thus, it bridges the gap between classical risk assessment techniques like fault trees or reliability block diagrams, allowing early feedback in the system design phase. However, the goal is not a final verification, as here the confidence also needs consideration. The main purpose is for early design stages and a low amount of available test data. In Section 3, a simulation study is performed to validate the usefulness of the bound. Additionally, the use is demonstrated by the example of breast cancer prediction. Section 4 elaborates on limitations and possible future work before Section 5 concludes the paper.

2. Method

The classification model will be denoted with f, and is a function from the input space $\mathbb{R}^d$ into the labels $\{1, \dots, K\}$, with $d \in \mathbb{N}_{>0}$ For simplicity, we decided to keep numbers for the labels, even though no ordinal scale is assumed. Additionally, f is given via a score function $s \colon \mathbb{R}^d \to \mathbb{R}^K$ by

$$f(x) = \underset{j=1,\dots,K}{\operatorname{argmax}} s(x)_j . \tag{1}$$

In simple words, s represents a belief of the model that x lies in each of the classes. What might look like an assumption is just a mathematical formulation, as any black-box classifier can be brought into this form. All occurring functions are assumed to be measurable, in order to soundly talk about probabilities.

The test data is given by the feature vector and its corresponding label, in example $(x_1, y_1), \dots, (x_N, y_N) \in \mathbb{R}^d \times \{1, \dots, K\}$. To include also aleatoric uncertainty, the data is viewed as realizations of the random variables $(X_1, Y_1), \dots, (X_N, Y_N)$. In particular $(x_j, y_j) = (X_j(\omega_0), Y_j(\omega_0))$, with $\omega_0 \in \Omega$ being the observed stochastic event from the underlying probability space $(\Omega, \mathcal{A}, P)$. A new unseen data point is written without a subindex as (X, Y). Model training and retrieval are disregarded, and only testing and evaluation are in focus; therefore, we only consider test data.

With $\mathcal{U}(y)$ we denote the set of undesired outcomes for a prediction, when the true label is y. If we are in the case of binary classification, so $K = 2$, then $\mathcal{U}(1) = \{2\}$. This distinction differs from model training and performance measures, as some failure modes are dropped. Even though this looks more complicated than treating every misclassification equally important, it allows us to exclude cases that are not safety relevant. To give an example, if an object detector of an autonomous vehicle detects a deer on the road instead of a boar, this failure might not influence the behaviour of the vehicle controller and is therefore not safety relevant.

Before presenting the method, we present our main restriction, which gives rise to the title of this paper.

Assumption 2.1. *The data is independent and identically distributed with*

$$Z = \max_{j \notin \mathcal{U}(Y)} s(X)_j - \max_{j \in \mathcal{U}(Y)} s(X)_j , \tag{2}$$

following a normal distribution of mean μ and variance σ^2.

2.1. *Bound Computation and Confidence Selection*

The value of interest is given by

$$\mathcal{R}(f) = P(f(X) \in \mathcal{U}(Y)), \tag{3}$$

which we call the risk of the classifier. It describes the probability that the model produces an undesired output on a single prediction for a new data point. According to (1), this occurs when the score of an undesired class is higher, respectively equal, since in this case the behaviour is unknown unless further specified, than all other scores from the desired labels. In other words, the variable Z, given in Equation (2) is 0 or smaller, thus

$$\mathcal{R}(f) = P(Z \leq 0) \overset{2.1}{=} \Phi\left(-\frac{\mu}{\sigma}\right). \tag{4}$$

Now, the first problem occurs, namely, the parameters μ and σ are unknown. Thus, we will use the commonly known estimators

$$\bar{\mu}_N(\omega) = \frac{1}{N}\sum_{j=1}^{N} Z_j(\omega), \tag{5}$$

$$\bar{\sigma}_N(\omega) = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N}\left(Z_j(\omega) - \bar{\mu}_N(\omega)\right)^2}, \tag{6}$$

for $\omega \in \Omega$ being a stochastic event. Using the estimators directly does not protect for possible under- or overestimation. Thus, we are going to use the following well-known confidence bounds,

$$P\left(\bar{\mu}_N - F^{-1}_{t_{N-1}}(1-\gamma)\frac{\bar{\sigma}_N}{\sqrt{N}} \leq \mu\right) = 1-\gamma, \tag{7}$$

$$P\left(\sigma^2 \leq \frac{(N-1)\bar{\sigma}_N^2}{F^{-1}_{\chi^2_{N-1}}(\eta)}\right) = 1-\eta, \tag{8}$$

where $\gamma, \eta \in (0,1)$ are the confidence levels and F^{-1} represents the inverse cumulative distribution function of a student t-distribution, respectively χ^2-distribution, with $N-1$ degree of freedom. We will write M for the stochastic event inside the argument in Equation 7 and V for the stochastic event inside the probability in Equation 8.

To obtain an upper bound estimation, we are going to distinguish between two scenarios. Namely that ω lies in $M \cap V$ or not, resulting in

$$\mathcal{R}(f) = \mathcal{R}(f)\cdot \mathbb{1}_{M\cap V}(\omega) \tag{9}$$

$$+ \mathcal{R}(f)\cdot \mathbb{1}_{\Omega\setminus M\cap V}(\omega) \tag{10}$$

$$\leq \Phi\left(\underbrace{\frac{-\bar{\mu}_N(\omega) + F^{-1}_{t_{N-1}}(1-\gamma)\frac{\bar{\sigma}_N(\omega)}{\sqrt{N}}}{\bar{\sigma}_N(\omega)\sqrt{\frac{N-1}{F^{-1}_{\chi^2_{N-1}}(\eta)}}}}_{\nu(\gamma,\eta)(\omega)}\right) \tag{11}$$

$$\cdot \mathbb{1}_{M\cap V}(\omega) + 1\cdot \mathbb{1}_{\Omega\setminus M\cap V}(\omega). \tag{12}$$

As noticed by the attentive reader, we additionally need the condition $-\frac{\bar{\mu}_N(\omega)}{\bar{\sigma}_N(\omega)} + \frac{F^{-1}_{t_{N-1}}(1-\gamma)}{\sqrt{N}} < 0$ to hold true for all $\omega \in M \cap V$ in order for the estimation to hold true. This is, of course, impossible to validate, since only the event ω_0 is observed, thus we can only check the condition at the measured test data. However, with increasing N, the last summand will converge to zero for fixed γ. Furthermore, since the estimator μ_N also converges to μ almost everywhere, this condition can be viewed as a technical adaption of $\mu > 0$, which according to Equation (4) relates to the question of $\mathcal{R}(f) < 0.5$, so if the classifier is worse than chance in this regard. Next we apply $\mathbb{1}_{M\cap V} \leq 1$ and then the expected value to get

$$\mathcal{R}(f) \leq \mathbb{E}\left[\Phi(\nu(\gamma,\eta))\right] + P(\Omega \setminus M \cap V). \tag{13}$$

Finally, to make any use of this bound, we need an approximation of the expected value in Equation (13). However, our only measurement is the observations of the data at ω_0, thus we use

$$\mathbb{E}\left[\Phi(\nu(\gamma,\eta))\right] \approx \Phi(\nu(\eta,\gamma)(\omega_0)). \tag{14}$$

Additionally, the second summand in Equation (13) can be upper bounded by the sum of $1 - P(M) = \gamma$ and $1 - P(V) = \eta$ due to Equations (7) and (8). This results in the estimated upper

bound

$$\mathcal{R}(f) \lessapprox \Phi\left(\frac{\frac{-\bar{\mu}_N(\omega_0)}{\bar{\sigma}_N(\omega_0)} + \frac{F^{-1}_{t_{N-1}}(1-\gamma)}{\sqrt{N}}}{\sqrt{\frac{N-1}{F^{-1}_{\chi^2_{N-1}}(\eta)}}}\right) + \gamma + \eta \tag{15}$$

$$= g_N(\gamma, \eta). \tag{16}$$

Highly interesting is the fact that the risk is independent of the confidence levels γ and η, while g_N has these as its arguments. Therefore we can simply take the minimizing γ_{mi} and η_{mi}. This also directly gives us choices for the confidence levels of the parameters.

2.2. *Computation Aspects*

The upper bound function is differentiable, as given below in Theorem 2.1.

Theorem 2.1. *The partial derivatives of g_N are given by*

$$\frac{\partial g_N}{\partial \gamma}(\gamma_0, \eta_0) = -\frac{1}{\sqrt{2}} \exp\left(-\frac{\nu(\gamma_0, \eta_0)^2}{2}\right) \tag{17}$$

$$\cdot \sqrt{\frac{F^{-1}_{\chi^2_{N-1}}(\eta_0)}{N}} \cdot \frac{\Gamma\left(\frac{N-1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \tag{18}$$

$$\cdot \left(1 + \frac{F^{-1}_{t_{N-1}}(1-\gamma_0)^2}{N-1}\right)^{\frac{N}{2}} + 1, \tag{19}$$

$$\frac{\partial g_N}{\partial \eta}(\gamma_0, \eta_0) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\nu(\gamma_0, \eta_0)^2}{2}\right) \tag{20}$$

$$\cdot \frac{2^{\frac{N-3}{2}}\Gamma\left(\frac{N-1}{2}\right)}{\sqrt{N-1}} F^{-1}_{\chi^2_{N-1}}(\eta_0)^{1-\frac{N}{2}} \tag{21}$$

$$\cdot \left(-\frac{\bar{\mu}_N}{\bar{\sigma}_N} + \frac{F^{-1}_{t_{N-1}}(1-\gamma_0)}{\sqrt{N}}\right) \tag{22}$$

$$\cdot \exp\left(\frac{F^{-1}_{\chi^2_{N-1}}(\eta_0)}{2}\right) + 1. \tag{23}$$

Thus, a simple gradient descent method can be applied to find the minimum. We decided on an iterative minimization in each variable, as displayed in Algorithm 1. However, for large values of N,

Algorithm 1 Heuristic to find the minimum of g.

Require: $\bar{\mu}_N > 0, \bar{\sigma}_N > 0, N > 0, \varepsilon > 0$

$\gamma \leftarrow \frac{1}{4}$ ▷ Initialize γ
$\eta \leftarrow \frac{1}{4}$ ▷ Initialize η
$m \leftarrow g(\gamma, \eta)$ ▷ Initialize minimum
$\Delta \leftarrow \infty$ ▷ Variable for current change
repeat
  $\gamma \leftarrow \operatorname{argmin}_{\tilde{\gamma} \in (0,1)} g_N(\tilde{\gamma}, \eta)$ ▷ Minimize in γ
  $\eta \leftarrow \operatorname{argmin}_{\tilde{\eta} \in (0,1)} g_N(\gamma, \tilde{\eta})$ ▷ Minimize in η
  $\Delta \leftarrow |m - g_N(\gamma, \eta)|$ ▷ Update change
  $m \leftarrow g_N(\gamma, \eta)$ ▷ Update minimum
until $\Delta < \varepsilon$
if $-\frac{\bar{\mu}_N}{\bar{\sigma}_N} + \frac{F^{-1}_{t_{N-1}}(1-\gamma)}{\sqrt{N}} < 0$ **then**
  return m
else
  return 1
end if

the terms

$$\frac{\Gamma\left(\frac{N-1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \quad \text{and} \quad \frac{2^{\frac{N-3}{2}}\Gamma\left(\frac{N-1}{2}\right) F^{-1}_{\chi^2_{N-1}}(\eta)^{1-\frac{N}{2}}}{\sqrt{N-1}} \tag{24}$$

cause numerical difficulties. The first term is of the form "$\frac{\infty}{\infty}$" for large N, but can be approximated with $\sqrt{\frac{2}{N-1}}$ for large N according to Wendel (1948); Brain and Mi* (2001). The second faces the problem of "$0 \cdot \infty$" for large N and is not as easy to solve; therefore, we propose a simple stepwise and directed line search in cases where N is too large. In this way, we do not need to access the derivative.

2.3. *Refined Failure Estimates*

Despite the flexibility of the definition in Equation 3, in practice, it might be necessary to obtain quantification for more refined failure modes. For instance, a bound for misclassifying a data point from class c_t to class c_f may be required in the risk assessment. This can also be done with the

framework above, only by a slight modification. Therefore, we assume that the labels Y_j are almost surely constant, in other words, all labels are constant on a set with probability 1. Ignoring the case of probability 0, this allows us to assign each data sample to one class, which was not possible before. Moreover, we need all points belonging to the same class to satisfy Assumption 2.1. Consequently, the value of interest is

$$P\left(f\left(X\right) \in U\left(Y\right) \mid Y = c_t\right), \tag{25}$$

with $\mathcal{U}\left(c_t\right) = \{c_f\}$. So if we discard all test points that do not have labels c_t, and assume $Y = c_t$, the condition in probability can be dropped and the framework applied in the above manner. Please note that the assumptions for this specific failure mode are fundamentally different to the general case and can not hold simultaneously. In the next Section 3, we also included these class-specific risks to provide a practical example in this regard.

3. Validation

The purpose of this section is to first evaluate the overall confidence of the previous bound estimation and secondly present an example of usage in practice. All computations have been carried out in Matlab R2025b, the code can be found online[a].

3.1. *Confidence Evaluation*

The goal of this subsection is to obtain an estimation of the confidence of the estimated bound. In particular, we are interested in the frequency with which our bound exceeds the true risk. To compute this proportion, we simulated the Z variables directly. In particular for each $N \in \{100, 102, 104, \ldots, 300\}$, $\mu \in \{0.01, 0.03, 0.05, \ldots, 1.99\}$ and $\sigma \in \{0.01, 0.03, 0.05, \ldots, 1.99\}$ we simulated N normal realization of mean μ and standard deviation σ. Afterwards, we computed the bound and compared the results to the true risk given via Equation (4). For theoretical correctness towards the confidence terminology, this procedure should have been repeated, however due to resource limitations, this would have caused a dramatic reduction of parameters. Therefore, we decided on the presented choice without repetitions.

As a result, in 216 cases our proposed bound was incorrect, which corresponds to a fraction of 2.1386×10^{-4} invalid bounds. These results emphasise a good reliability for the intended purpose.

3.2. *Academic Example*

The purpose of this little example is to demonstrate the need for a margin in error estimation. To stay as simple as possible, we consider binary classification where the data from class 1 is normal distributed with mean μ_1 and standard deviation σ_1 as

$$\mu_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \sigma_1 = \begin{pmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{pmatrix}. \tag{26}$$

Class 2 is also normal distributed with mean μ_2 and standard deviation σ_2 as

$$\mu_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0.35 & 0.1 \\ 0.1 & 0.25 \end{pmatrix}. \tag{27}$$

Please note that we are only considering the specific classes, if the corresponding overall variables Z are normal, identical distributed or independent is unknown at first. As a model we use a linear support vector machine, which can be represented as $1 + H\left(\langle w, x\rangle + b\right)$ for H being the Heaviside step function, $\langle \cdot, \cdot \rangle$ being the euclidean scalar product, and $w \in \mathbb{R}^d$, $b \in \mathbb{R}$ be the parameters. Consequently, the score $s\left(x\right) = \langle w, x\rangle + b$ is also normal distributed. Please note that the case of score 0 is not specified in MATLAB. Additionally, the classes appear to have inverted scores, which would result in $2s$, so still normal distributed.

In our experiment, we simulated 1000 data points for training and $N \in \{150, 200, 250\}$ data points for testing, with an equal split between the classes. A scatter plot for the $N = 150$ case is printed in Figure 1. Afterwards, we computed the error rate for each class, as displayed in Table 1. Additionally, the direct use of estimators, in example $\Phi\left(-\frac{\bar{\mu}_N}{\bar{\sigma}_N}\right)$ is also printed. As can be seen, measured failure rates on the test sets are above these values, indicating that a direct use of

[a] https://github.com/Gueni96/Single-valued-Risk-Bound-for-Normal-Classifiers

estimators in the error modelling can be incorrect. This directly stems from the fact that the estimators depend on the observed event ω_0.

The bound as described in Section 2 values are also printed in Table 1 and we can see that they are much higher than the actual performance. However, this is the behaviour to expect, since the performance is only measured on $N = 75, 100, 125$ samples each, which alone makes the precision at most $0.013, 0.01, 0.008$. Therefore, a high amount of uncertainty is present and captured within our framework. If the number N increases, the bound decreases, as it can be seen in class 1. The bound at class 2 has some mixed behaviour, possibly stemming from the increase in the failure rate.

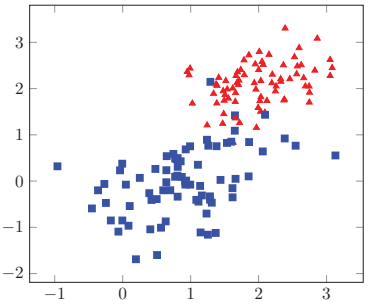

Fig. 1. Scatter Plot of the test data in the academic example for $N = 150$.

3.3. *Breast Cancer Prediction*

In this subsection, we are going to demonstrate our method at the example of breast cancer prediction. We are going to use the data set published at Wolberg (1990) because it also allows us to point out some pitfalls and aspects to keep track of.

We started by removing samples with missing values, which left 683 data points remaining. Afterwards, the data has been randomly split into 383 samples for training and 300 for testing. As a model, a linear support vector machine was used.

Before deploying our method, the preconditions must be evaluated. One of them is independence, which is difficult to guarantee in practice. In our case, according to Mangasarian and Wolberg (1990), the data set contains duplicates and therefore does not fulfil the requirement. We still decided to choose this data set to emphasise that this condition must also be considered, and if necessary, switched to alternative frameworks like Günther et al. (2025), that allow dependencies. However, we will continue as if independence had been fulfilled. Next, the condition of identical distribution must be considered. According to Wolberg and Mangasarian (1990), the numerical feature values can be reproduced by different observers. This alone does not ensure identical distribution but indicates some stability, therefore we are also setting a check mark to this property. Finally, the requirement of normal distribution needs to be judged. There are different possibilities to do so; one is to perform statistical tests. So did we, and as a result, only the Z variables, restricted to the malignant class, passed the Anderson-Darling test and Jarque-Bera test at 5% significance level. We also included the histogram plots in Figure 2, to show that a visual sanity check can be hard to judge objectively.

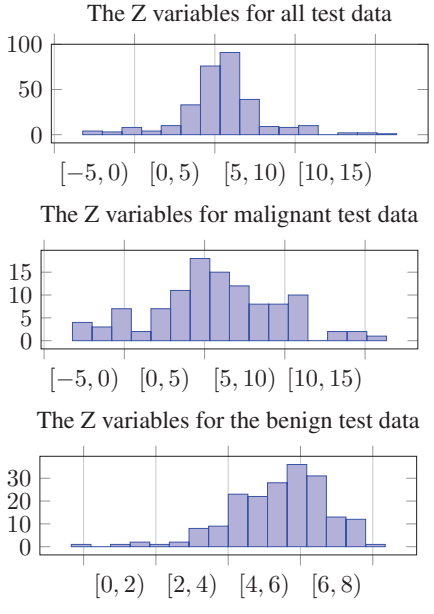

Fig. 2. Histogram of the different test data choices.

So we can continue and compute an upper bound estimation for the malignant class. There-

Table 1. Failure rate estimations and bounds of the academic example. The mantissa of the result is rounded to 2 places.

N	Class 1: Failure Rate	Class 1: $\Phi\left(-\frac{\bar{\mu}_N}{\bar{\sigma}_N}\right)$	Class 1: $g_N\left(\gamma_{\text{mi}}, \eta_{\text{mi}}\right)$	Class 2: Failure Rate	Class 2: $\Phi\left(-\frac{\bar{\mu}_N}{\bar{\sigma}_N}\right)$	Class 2: $g_N\left(\gamma_{\text{mi}}, \eta_{\text{mi}}\right)$
150	4×10^{-2}	5.95×10^{-2}	17.25×10^{-2}	1.33×10^{-2}	1.61×10^{-2}	8.40×10^{-2}
200	2×10^{-2}	5.40×10^{-2}	14.59×10^{-2}	2×10^{-2}	4.21×10^{-2}	12.47×10^{-2}
250	5.6×10^{-2}	5.28×10^{-2}	13.34×10^{-2}	5.6×10^{-2}	3.70×10^{-2}	10.64×10^{-2}

fore, we minimize the function g_N as described in subsection 2.2. A plot of g_N can be found in Figure 3. As can be seen, g_N for the malignant

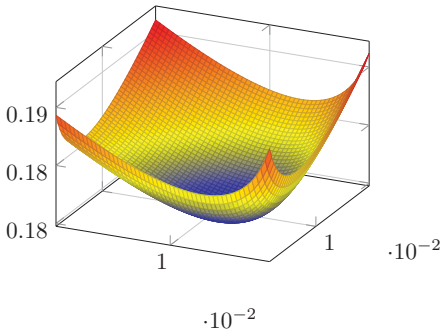

Fig. 3. Surface plot of g_N for the breast cancer prediction failure of classifying malignant to benign.

class has a clear local minimum.

Despite the fact that the other failures are not normal distributed, we computed the bounds and presented them in Table 2. As can be seen, if the normal assumption is not fulfilled, the bounds can become invalid, therefore it is important to validate the prerequisites. As we observe in the malignant class, our bound is paying tribute to the low number of test cases, namely 110 test cases.

4. Limitations and Future Work

The main limitation is given by the Assumption 2.1, and it is currently unclear how well the data must represent the distribution. As used in Section 3, tests for normal distribution exist, but how good the approximation must be to provide reliable results remains a research question. Future work could therefore tackle this problem via an extended simulation. Another related problem is presented via the identical distribution, so differences in the test data and the actual application. In order for the framework to work, those must coincide. Possible solutions are the use of additional wrappers like Kläs and Sembach (2019) or confidence scores Ghobrial et al. (2023) to capture outliers during runtime.

5. Conclusion

In this work, we have presented one possibility to compute an upper bound estimation for the failure on demand, in the case of a normal classifier. Hereby, an upper bound function is built that depends on the confidence levels of the parameter estimators. Minimizing this function results in an upper bound estimation and a determination of optimal confidence levels. The overall confidence is determined via a simulation, and the application of the framework is presented with the example of breast cancer prediction. This will allow a follow-up risk assessment in early design stages, even if the number of test samples is small.

References

Bousquet, O., S. Boucheron, and G. Lugosi (2004). Introduction to statistical learning theory. In O. Bousquet, U. von Luxburg, and G. Rätsch (Eds.), *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2003, Tübingen, Germany, August 2003, Revised Lectures*, pp. 169–207. Berlin, Heidelberg: Springer Berlin Heidelberg.

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

Brain, C. W. and J. Mi* (2001). On some prop-

Table 2. Failure rate estimations and bounds of the breast cancer prediction.

Measure	Overall	Class Benign	Class Malignant
Failure Rate	3.67×10^{-2}	5.26×10^{-3}	9.09×10^{-2}
$\Phi\left(-\frac{\bar{\mu}_N}{\bar{\sigma}_N}\right)$	1.90×10^{-2}	3.52×10^{-5}	7.79×10^{-2}
$g_N\left(\gamma_{\text{mi}}, \eta_{\text{mi}}\right)$	5.06×10^{-2}	1.53×10^{-3}	1.76×10^{-1}

erties of the quantiles of the chi-square distribution and their applications to interval estimation. *Communications in Statistics - Theory and Methods 30*(8-9), 1851–1867.

Chilenski, J. J. and S. P. Miller (1994). Applicability of modified condition/decision coverage to software testing. *Software Engineering Journal 9*, 193–200.

Dijkman, J., H. van Haeringen, and S. de Lange (1983). Fuzzy numbers. *Journal of Mathematical Analysis and Applications 92*(2), 301–341.

Esteva, A., B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun (2017, Feb). Dermatologist-level classification of skin cancer with deep neural networks. *Nature 542*(7639), 115–118.

Ghobrial, A., D. Hond, H. Asgari, and K. Eder (2023). A trustworthiness score to evaluate dnn predictions. In *2023 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pp. 9–16.

Günther, A., S. Vollmer, and P. Liggesmeyer (2025). Single-valued risk estimation in classification for dependent data. In *2025 IEEE International Conference on Artificial Intelligence Testing (AITest)*, pp. 24–33.

Kabir, S. and Y. Papadopoulos (2018). A review of applications of fuzzy sets to safety and reliability engineering. *International Journal of Approximate Reasoning 100*, 29–55.

Kläs, M. and L. Sembach (2019). Uncertainty wrappers for data-driven models. In A. Romanovsky, E. Troubitsyna, I. Gashi, E. Schoitsch, and F. Bitsch (Eds.), *Computer Safety, Reliability, and Security*, Cham, pp. 358–364. Springer International Publishing.

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.

Pei, K., Y. Cao, J. Yang, and S. Jana (2019, October). Deepxplore: automated whitebox testing of deep learning systems. *Commun. ACM 62*(11), 137–145.

Pilch, M., T. G. Trucano, and J. C. Helton (2011). Ideas underlying the quantification of margins and uncertainties. *Reliability Engineering & System Safety 96*(9), 965–975. Quantification of Margins and Uncertainties.

Sun, Y., X. Huang, D. Kroening, J. Sharp, M. Hill, and R. Ashmore (2019, October). Structural test coverage criteria for deep neural networks. *ACM Trans. Embed. Comput. Syst. 18*(5s), 94:1–94:23.

Wendel, J. G. (1948). Note on the gamma function. *The American Mathematical Monthly 55*(9), 563–564.

Wolberg, W. (1990). Breast Cancer Wisconsin (Original). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5HP4Z.

Wolberg, W. H. and O. L. Mangasarian (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences 87*(23), 9193–9196.

Zadeh, L. (1965). Fuzzy sets. *Information and Control 8*(3), 338–353.

Zadeh, L. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems 100*, 9–34.