ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.30, No-1-2, 2022

Uncertainty-aware Evaluation of Machine Learning Performance in binary Classification Tasks

Leo Sperling
University of Kaiserslautern
Gottlieb-Daimler-Strasse 47
Germany, 67663
Kaiserslautern, RLP
sperling@rhrk.uni-kl.de

Simon Lämmer
Leipzig University
Ritterstr. 26
Germany, 04109
Leipzig, Saxony
simon.laemmer@outlook.de

Hans Hagen
University of Kaiserslautern
Gottlieb-Daimler-Strasse 47
Germany, 67663
Kaiserslautern, RLP
hagen@cs.uni-kl.de

Gerik Scheuermann
Leipzig University
Ritterstr. 26
Germany, 04109
Leipzig, Saxony
scheuermann@informatik.uni-leipzig.de

Christina Gillmann
Leipzig University
Ritterstr. 26
Germany, 04109
Leipzig, Saxony
gillmann@informatik.uni-leipzig.de

ABSTRACT

Machine learning has become a standard tool in computer vision. Nowadays, neural networks are one of the most prominent representatives in this class of algorithms that usually require training and evaluation to work as desired. There exist a variety of evaluation metrics to determine the quality of a trained neural network, which are usually threshold dependent. This results in massive changes in the resulting evaluation when the threshold is changed slightly. Further, measurements of uncertainty such as resulting from Bayesian approaches, are not considered in this analysis. In this paper, we present evaluation metrics for machine learning approaches that are able to attach a probability distribution to the utilized threshold and include uncertainty measures. We demonstrate the applicability of our approach by applying the defined metrics to a real-world example where a Bayesian neural network has been used to predict stroke lesions.

Keywords

Evaluation Measures, Uncertainty-awareness, Machine Learning

1 INTRODUCTION

Machine learning approaches become increasingly important in the area of computer vision [1]. Especially in classification tasks, machine learning approaches have developed into a standard tool, massively reshaping the respective area. In this process, the evaluation of machine learning approaches is a crucial factor. Here, a variety of measures exist that aim to examine the performance using different assumptions and focus points.

As input data, models, and the use of visualization usually include uncertainty [2], the evaluation of machine learning approaches can be affected. Sacha et al. [3] proposed, that uncertainty has a crucial impact on the decision-making process. Unfortunately, the existing measures do not include uncertainty in their computation.

Evaluation measures such as DICE-coefficient (=F1 score) and accuracy for machine learning approaches, usually do not consider the uncertainty inherent in the machine learning process. Instead, they consider a pre-selected threshold and build their computation based on true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Unfortunately, the selection of this threshold holds a large potential of uncertainty. Slight changes in the choice of the threshold can have a massive impact on the resulting evaluation.

In addition, fuzzy machine learning approaches, such as Bayesian Neural Networks [4], output a measure of uncertainty in addition to the classification prediction, which is usually not considered in the evaluation of machine learning approaches. This results in an evaluation that is equally balanced along all classifications made in

a machine learning model, independent of how certain the prediction is (see Section 2).

In this work, we aim to revisit popular evaluation measures for machine learning approaches that target binary classification tasks (see Section 3). To achieve this, we first rephrase the terms TP, TN, FP, FN, such that we use an uncertainty-aware threshold. Based on this, we can rebuild popular machine learning evaluation measures to include the uncertainty-aware threshold. Further, we include a damping factor that allows adjusting the importance of predicted classifications based on measured uncertainty.

Therefore, this paper contributes:

- Uncertainty-aware classification of machine learning results
- A mechanism to include uncertain classification results in machine learning evaluation
- Uncertainty-aware evaluation measures for machine learning approaches

We show how the defined uncertainty-aware measures can be used in machine learning performance evaluation using varying examples as shown in Section 4. Our results will be discussed in Section 5.

2 RELATED WORK

In the context of the presented approach, we aim to analyze previous work conducted in the area of uncertainty-aware machine learning and the evaluation of these approaches.

2.1 Uncertainty-aware Machine Learning

The importance of uncertainty analysis in the area of machine learning has been highlighted by Klaes et al. [5]. In their work, they summarize potential sources of uncertainty in the respective area. The presented taxonomy holds a valuable starting point in the presented area of research. This approach was also refined for machine learning approaches in medical imaging [6]. The described sources of uncertainty are manifold and therefore various approaches exist that aim to target one or multiple sources of uncertainty.

Sluijterman et al. [7] provided an adapted approach of regression, that aims to include uncertainty quantification during the computation. Nieradzik et al. [8] exchanged the output activation function which is usually set to the sigmoid function with further functions and examined their suitability regarding the resulting prediction and their uncertainty. Ding et al. [9] proposed an uncertainty-aware training, where training data is adapted such that more reliable data points become more important in the training process. Eldesokey et al. [10] aimed for a holistic uncertainty-aware machine learning approach that includes multiple sources of uncertainty. Here, uncertainty arising from the data as well as the uncertainty of the model is included throughout the entire computation of the machine learning approach. Although these approaches all target the incorporation of uncertainty into the training process, they rely on the classic evaluation approaches for machine learning approaches, which are threshold-based. In this work, we aim to extend these approaches such that the threshold holds a probability distribution function to indicate its potential uncertainty.

Recently, the number of machine learning approaches that explicitly work with mathematical concepts that directly include uncertainty increased. Here, approaches such as fuzzy deep networks [11] or Bayesian neural networks [4] that can output epistemic and aleatoric uncertainty [12] in their prediction have been developed. Epistemic uncertainty refers to uncertainty inherent in a model, as models are always making assumptions. On the other hand, aleatoric uncertainty refers to uncertainty inherent in captured data due to random effects and measurement imprecision.

Also, Sacco et al. [13] proposed a neural network approach that builds a second neural network to predict the uncertainty inherent in the computational process. All these approaches are able to attach an uncertainty to the made prediction. Still, these values are usually only reviewed visually but are not considered in the evaluation of the proposed approach. In this work, we aim to provide a mechanism to include this knowledge.

2.2 Uncertainty-aware evaluation of Machine Learning

The evaluation of machine learning approaches is a key point while using them. There exist a variety of surveys and books that summarize and categorize them [14, 15]. These measures include DICE-coefficient, accuracy, recall, and precision and are used for benchmarking [16]. Their selection is dependent on the underlying problem and type of used machine learning approach [17]. All these measures are based on the separation of predicted values into TP, TN, FP, and FN. Here, a threshold is selected to achieve this separation. The choice of this threshold can have a massive influence on the evaluation of the machine learning approach and needs to be adjusted in each case.

Gao et al. [18], presented an approach that aims to generate a self-adapting threshold for the evaluation of a neural network. The method is built on an analysis of the imbalance of classes that are predicted. Thada et al. [19] adapted the threshold for evaluation based on the underlying scale of predicted classifications. Here, different scales obtain different thresholds. Li et al. [20] provided a machine learning approach that aims

to guess a proper threshold based on the underlying dataset. Here, different thresholds are examined to understand the resulting classification. Although these approaches aim to select the threshold that is used for evaluation, the choice may still remain uncertain. Therefore, our approach aims to add a probability distribution function to the selected threshold to express the uncertainty in this decision.

Taha et al. [21] presented a set of evaluation metrics that are based on fuzzy theory. Here, the prediction and the ground truths are considered as fuzzy sets, and metrics are presented that compare them. Although this gives a valuable starting point for the presented work, the approach is not able to indicate an uncertainty-aware threshold. In addition, the inclusion of uncertainty that can result from a neural network cannot be included in this approach.

Psaros et al. [22] provided evaluation metrics that aim to include the uncertainty that can be outputted by machine learning approaches. Here, prominent metrics are adapted individually to include uncertainty information. Still, this approach is based on a fixed threshold. In the presented approach we aim to present a generalized way to include an uncertainty-aware threshold as well as a damping factor that adjusts made classifications based on the underlying uncertainty.

3 METHODS

To achieve uncertainty-aware evaluation measures for machine learning, we first aim to extend prominent measures of neural network performance to include a probability distribution to the user-defined threshold. Based on this, we will further include potential uncertainty measures that can be outputted by Bayesian Network approaches.

3.1 Uncertainty-aware classification

Neural Networks aim to learn from existing datasets. To test the performance of the neural network, the predicted results are compared to a ground truth. In general, the closer both are to each other, the better the performance. Here, each datapoint i, and its classification $c(i)$ is compared to the prediction $p(i)$. Note that we restrict the range of $c(i)$ to 1 and 0, while the range of $p(i)$ is the interval $[0,1]$, as most machine learning approaches output probabilities instead of fixed class assignments.

Most evaluation measures for neural networks work based on a classification of values into TP, FP, TN, FN, which are based on a pre-selected threshold t. Therefore, the following definitions are known:

$$TP_i(t) = \begin{cases} \overbrace{1}^{A} & \overbrace{c(i)}^{B} \wedge \overbrace{[p(i) \geq t]}^{C} \\ 0 & else \end{cases} \tag{1}$$

$$TN_i(t) = \begin{cases} 1 & !c(i) \wedge [p(i) \leq t] \\ 0 & else \end{cases} \tag{2}$$

$$FP_i(t) = \begin{cases} 1 & !c(i) \wedge [p(i) > t] \\ 0 & else \end{cases} \tag{3}$$

$$FN_i(t) = \begin{cases} 1 & c(i) \wedge [p(i) < t] \\ 0 & else \end{cases} \tag{4}$$

with respect to the threshold t. We define subequations for future reference in this manuscript to allow easy to follow changes that we make. A is defined as the function output of the classification functions. B represents the classification that was made by a neural network and C represents the groundtruth that is is used for the comparison.

By definition, the result of these functions can only be 0 or 1. A is either 0 or 1 in a fixed case. In our approach, we also consider uncertain ground truths. Although most of the available training databases provide a fixed classification, the number of ground truths that hold fuzzy values increases. Therefore, we redefine $c(i)$ and allow it to lie in the range of $[0,1]$. Now, we also have to adjust the decision to which class a point belongs. Here, $B = [c(i) > t]$ holds. Still, we need to find a mechanism that allows rating the certainty of this decision.

Considering that we no longer work with fixed values of 1 and 0, we need to make an adaptation. We aim to define a general way to compare values to a threshold that has a probability distribution attached.

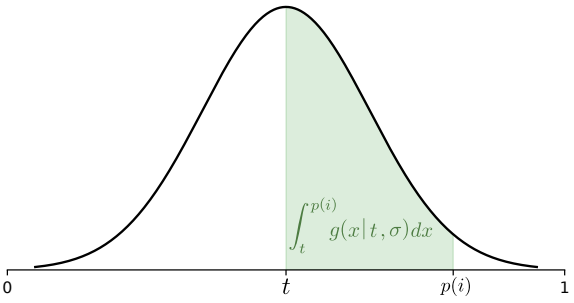


Figure 1: Schematic description of incorporation of Gaussian distribution of the threshold t.

As mentioned, the decision based on a fixed threshold results in fixed classifications. The decision to choose a threshold can be very hard as it is usually dependent on the underlying application. In addition, slight changes in the choice of the threshold can have a massive influence on the quality measures. Here, we use a Gaussian distribution function that is normalized:

$$g(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5}$$

A normalized Gaussian distribution function has beneficial attributes in the presented case. As the area under the curve is always 1, we can use this measure to adapt our previous classifications. Here, σ decides how sharp or flat the resulting Gaussian distribution is. We allow users to set the standard deviation σ in conjunction with the threshold t.

Here, a probabilistic measure if a threshold is exceeded can be expressed as:

$$\overline{A_{x_1}} = 2\int_{x_1}^{t} g(x|t,\sigma)dx \tag{6}$$

As the peak of the Gaussian distribution function is located at the threshold t, the maximum size of the area under the curve can be 0.5. As we aim for a measure that is located in the range of [0,1], we need to double this area. If t and x_1 are located close to each other, the resulting area under the curve converges to 0. A close location means a high uncertainty, which means that under this condition we aim to tone down the result of the classification. On the other hand, if the points are not close to each other, the area under the curve converges to 1. This results in low uncertainty. Resulting from this consideration, the classification scheme A can be rephrased as:

$$TP_i(t,\sigma)\begin{cases}\overline{A_{p(i)}}\cdot\overline{A_{c(i)}} & [c(i)>t]\wedge[p(i)>t]\\ 0 & else\end{cases} \tag{7}$$

The values of A computed in the measures $TN_i(t,\sigma)$, $FP_i(t,\sigma)$ and $FN_i(t,\sigma)$ are computed like this as well. Based on these extended definitions of TP, TN, FP, and FN, we further aim to include uncertainty that is captured in predictions by machine learning approaches.

3.2 Inclusion of predicted uncertainty

Recently, a variety of machine learning approaches is able to output uncertainty measures related to the made prediction. Here, especially Bayesian neural networks are able to output aleatoric as well as epistemic uncertainty measures. In this work, we aim to include these measures into the classifications. Here, we aim to achieve a weighting of the classification according to the outputted uncertainty measures. In particular, we aim for a classification scheme, that extends the existing scheme in the following manner $TP_i(t,\sigma,d) = TP_i(t,\sigma)\cdot C_d$, where C_d is supposed to work as a damping factor.

The goal of this factor is to tone down prediction values that are considered uncertain in the measures that can be outputted by uncertainty-aware machine learning approaches. We consider $u(i)$ as the uncertainty attached to the prediction value $p(i)$. Here, the uncertainty can be located in the range of $[0,\infty)$.

To define C_d, we aim for a function that outputs 1, if the uncertainty predicted by a machine learning approach is 0. In this case, the made classification will remain the same. In contrast, if a data point is classified as uncertain, we aim to let the damping function converge to 0. Here, we utilize the function:

$$C_d = e^{-(u(i)\cdot d)}, \tag{8}$$

where d works as an additional damping factor, that can be located in the range $[0,\infty)$.

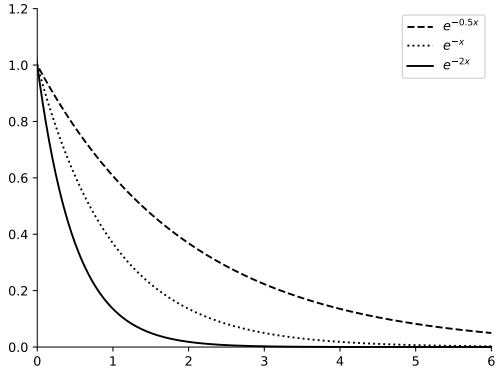


Figure 2: Different damping functions, based on the damping factor d. Examples are shown for $d = \frac{1}{2}$, $d = 1$, $d = 2$.

To indicate the effect of d, Figure 2 shows examples of interesting classes of the damping factor. The higher the damping factor, the more pronounced is the influence of the damping on the result. This gives the user a further input parameter with which to control the damping of the classification based on uncertainty quantification of the predictions made by machine learning approaches. If this quantification does not exist or cannot be achieved, C_d can be set to 1 and therefore does not change the made classifications.

Based on the made extensions of the classification schemes, we can extend the formulas from equations 1, 2, 4 and 3 using the scheme as explained:

$$TP_i(t,\sigma,d) = \begin{cases}TP_i(t,\sigma)\cdot C_d & [c(i)>t]\wedge[p(i)\geq t]\\ 0 & else\end{cases} \tag{9}$$

$$TN_i(t,\sigma,d) = \begin{cases}TN_i(t,\sigma)\cdot C_d & [c(i)\leq t]\wedge[p(i)\leq t]\\ 0 & else\end{cases} \tag{10}$$

$$FP_i(t,\sigma,d) = \begin{cases}FP_i(t,\sigma)\cdot C_d & [c(i)\leq t]\wedge[p(i)>t]\\ 0 & else\end{cases} \tag{11}$$

$$FN_i(t,\sigma,d) = \begin{cases} FN_i(t,\sigma)\cdot C_d & [c(i) > t] \wedge [p(i) < t] \\ 0 & else \end{cases} \tag{12}$$

Based on these definitions, we are able to extend well-known evaluation metrics for machine learning approaches.

3.3 Uncertainty-aware evaluation measures

In the following, we will summarize potential evaluation measures that are based on the prior classifications. The measures have been chosen as they are popular choices for machine learning evaluation [23]. Here, we need to sum all values that will be outputted when considering all n datapoints. Therefore we define $TP(t,\sigma,d) := \Sigma_{i=0}^{n} TP_i(t,\sigma,d)$. Respectively, $TN(t,\sigma,d)$, $FP(t,\sigma,d)$ and $FN(t,\sigma,d)$ can be defined.

$$\overline{Accuracy} := \frac{TP(t,\sigma,d)+TN(t,\sigma,d)}{TP(t,\sigma,d)+TN(t,\sigma,d)+FP(t,\sigma,d)+FN(t,\sigma,d)} \tag{13}$$

$$\overline{Precision} := \frac{TP(t,\sigma,d)}{TP(t,\sigma,d)+FP(t,\sigma,d)} \tag{14}$$

$$\overline{Recall} := \frac{TP(t,\sigma,d)}{TP(t,\sigma,d)+FN(t,\sigma,d)} \tag{15}$$

$$\overline{FalsePositiveRate} := \frac{FP(t,\sigma,d)}{FP(t,\sigma,d)+TN(t,\sigma,d)} \tag{16}$$

$$\overline{F1} := \frac{2\cdot\overline{Precision}\cdot\overline{Recall}}{\overline{Precision}+\overline{Recall}} \tag{17}$$

4 CASE STUDY

In this section, we aim to apply the developed uncertainty-aware evaluation metrics to a trained Bayesian U-Net (BNN) [24] for stroke lesion prediction [25]. We aim to show how the defined metrics can be used and how the defined parameters influence the computation.

4.1 Use Case Description

The provided BNN generates lesion maps from stroke patients that predict their final formation. Here, a multimodal input is used to predict a lesion map that can be found in the work of Gillmann et al. [26]. In addition, it predicts voxel-wise epistemic and heteroscedastic aleatoric uncertainty alongside [27]. The epistemic uncertainty stems from Monte Carlo dropout and is a property of the model used to describe the real-world process. It expresses not knowing exactly, which model generated the data in the real world. The heteroscedastic aleatoric uncertainty was trained as an unsupervised parameter in the loss function. We will use this model to demonstrate the applicability of the presented approach.

(a) (b) (c) (d) (e)

Figure 3: Cross section of lesion map prediction (a) and epistemic uncertainties (b) from BNN, thresholded at $t = 0.8$ (c) and compared to the ground truth (d). (e) shows the groundtruth (in red) overlayed on top of the CT Angiography, which is one of the inputs to the BNN.

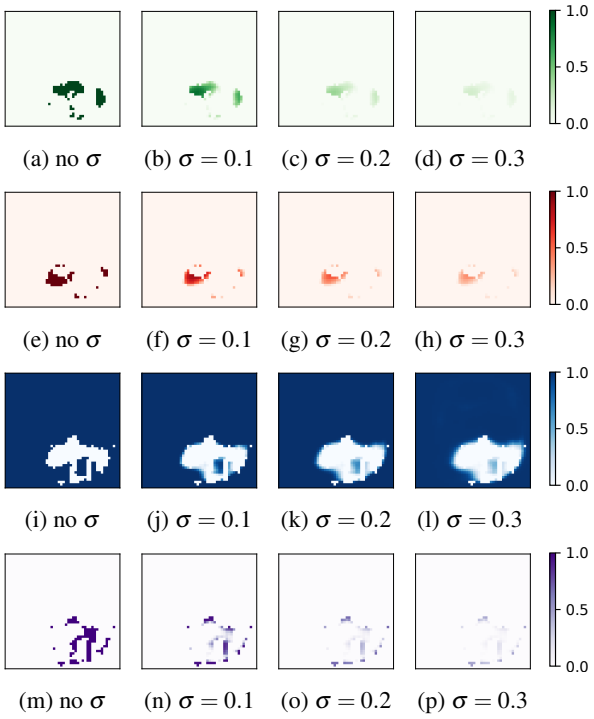
In this use case, we consider a particular cross-section of the 3D volume of a patient. Figure 3(a) shows the prediction made by the BNN, whereas 3(b) shows the predicted epistemic uncertainty. The prediction holds values between 0 (no lesion predicted) and 1 (lesion predicted). The ground truth that was labeled by medical experts is shown in Figure 3(d). Usually, performance measures are computed based on the thresholded prediction (Fig. 3(c)) and the pre-labeled ground truth. In this case, the groundtruth was created by medical experts that reviewed each patient individually and marked areas in the image that show a stroke lesion. For this example we show how the presented uncertainty-aware measures can be applied.

4.2 Results

In the following we aim to discuss the influence of the user-selected values σ and d to the classification values as well as the resulting metrics that can be computed based on these classifications. We define a consistent colorscheme for the four classes, i.e. TP (green), FP (red), TN (blue) and FN (purple).

Influence of σ to classifications As mentioned, the user-defined variability of the selected threshold shapes the sharpness of the classification result. The resulting classification of the presented prediction of stroke lesion into TP, FP, TN and FN with varying σ can be seen in Figure 4. 4(a), 4(e), 4(i) and 4(m) show the original computation of the classification. Here, clear boundaries can be identified due to the strict separation of classifiers. This coincides with an application of our evaluation approach when $\sigma \to 0$. Therefore, the presented measures extend the existing ones.

When increasing σ, the crisp boundaries of the classifications vanish and the separation into the classes is



(a) no σ (b) $\sigma = 0.1$ (c) $\sigma = 0.2$ (d) $\sigma = 0.3$

(e) no σ (f) $\sigma = 0.1$ (g) $\sigma = 0.2$ (h) $\sigma = 0.3$

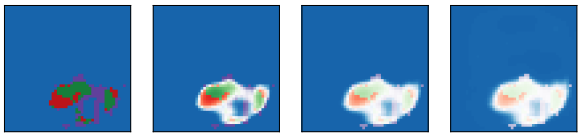
(i) no σ (j) $\sigma = 0.1$ (k) $\sigma = 0.2$ (l) $\sigma = 0.3$

(m) no σ (n) $\sigma = 0.1$ (o) $\sigma = 0.2$ (p) $\sigma = 0.3$

Figure 4: Classification of TP, FP, TN and FN (rows) with varying σ (columns). With larger σ, the score for classifications decreases according to the closeness to the threshold.

less clear. This matches with the intuition that a high σ represents a large uncertainty of the selected threshold.

Figure 5 shows the merged visualization of the made classification with varying σ. Figure 5(a) show the strict separation of the made prediction into the four classes. For increasing σ, an area with uncertain values is visible that indicates values of the prediction that are close to the threshold but would be considered as certain as the other predictions if the threshold is set fixed.



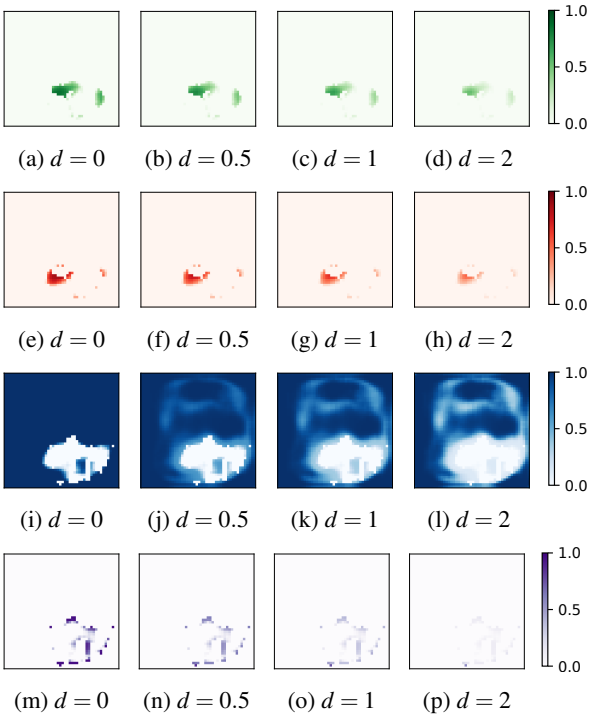
(a) no σ (b) $\sigma = 0.1$ (c) $\sigma = 0.2$ (d) $\sigma = 0.3$

Figure 5: Classification of TP, FP, TN and FN with different values for σ and fixed $d = 0$.

Influence of d The damping factor d has a large influence where the uncertainty of the BNN is high.

This effect to the classification metrics can be seen in Figure 6. Here, the value of σ is set to 0.1 in all cases. d is altered with $0, 0.5, 1$ and 2.

The damping factor controls the influence of the uncertainty on the made classifications. With increasing d, values that contain a high uncertainty will result in



(a) $d = 0$ (b) $d = 0.5$ (c) $d = 1$ (d) $d = 2$

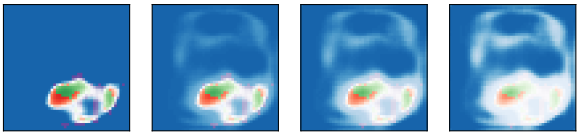
(e) $d = 0$ (f) $d = 0.5$ (g) $d = 1$ (h) $d = 2$

(i) $d = 0$ (j) $d = 0.5$ (k) $d = 1$ (l) $d = 2$

(m) $d = 0$ (n) $d = 0.5$ (o) $d = 1$ (p) $d = 2$

Figure 6: Classification of TP, FP, TN and FN (rows) for a fixed $\sigma = 0.1$ and varying d (columns).

a less strong classification. This effect can be seen very clearly when considering Figure 6(i) 6(j), 6(k) and 6(l). When setting d to 0, the result is almost binary. While increasing d, values with a high uncertainty get a lower classification score. When comparing the result of Figure 6(l) with Figure 3(b) we can identify the large influence of uncertain values in the prediction. Here, the uncertainty results in areas that cannot be separated clearly. Further, areas that do not contain a high uncertainty will not be affected by the application of the damping factor.



(a) $d = 0$ (b) $d = 0.5$ (c) $d = 1$ (d) $d = 2$

Figure 7: Classification of TP, FP, TN and FN with fixed $\sigma = 0.1$ and varying d.

The effect of varying the damping factor d on the combined image of the classifications into TP, FP, TN and FN is shown in Figure 7. Here, we can identify that a high uncertainty lowers the overall classification score of datapoints. When increasing d, uncertain areas will result in unclassified data values.

Influence on evaluation metrics Based on the made classifications, we adapted prominent examples of evaluation metrics.

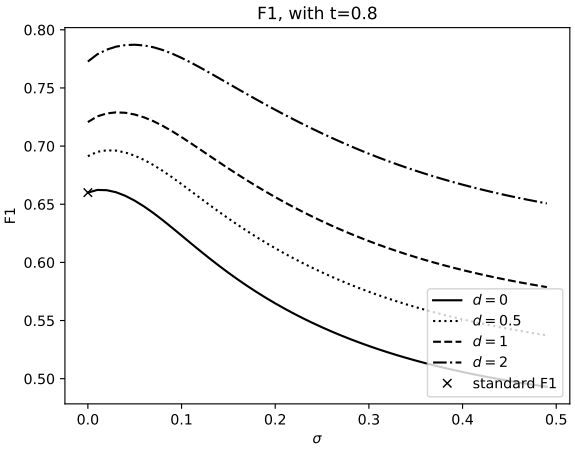


Figure 8: Results of adapted F1 metric. By incorporating the epistemic uncertainty information we get higher scores from the metric. The $\overline{F1}$ score increases for small σ, but for large σ we get lower scores.

Figure 8 shows the results of the adapted F1 metric (also known as DICE-coefficient) with a threshold of $t = 0.8$, variable σ and selected values for the damping factor d (0, 0.5, 1 and 2).

The unmodified F1 score is also highlighted in the graph (indicated with a black x). Incorporating the epistemic uncertainty into the classification by using a damping factor improved the final F1 score significantly while increasing the σ reduced the score overall. This results from the fact, that an increased σ removes confidence in the classifications and therefore lowers the result of the measurement output.

On the other hand, the damping factor removes uncertain values from the computation. Usually, these values are located around the boundary of areas in the ground truth which turn out to be classified wrongly in many cases. The damping factor removes these areas and therefore increases the overall performance result. At this point, we want to highlight that this effect might be reversed when uncertain areas are located within correctly predicted regions.

Interestingly, the best output of the evaluation metrics can be achieved with a sigma slightly lower than 0.1 and a damping factor of 2. In the given case this means that a consideration of an uncertainty-aware threshold leads to a better rating of the network performance. This fits with the intention of this work which aims to remove the fixed thresholding.

We also applied our adapted measures to further evaluation measures as shown in Figure 9. Here, we examined the measures $\overline{Accuracy}$, $\overline{Precision}$, $\overline{Recall}$ and $\overline{FPR}$.

When considering $\overline{Accuracy}$ (Figure 9(a)), we can identify that the unmodified accuracy metric shows a good result for the network (0.97). In the presented case, this is not surprising as the network predicts a high amount of TN correctly. Increasing σ results in even better ratings for the network as uncertain classifications are weighted less than certain classifications. In addition, an increased d further improves the network performance.

Figure 9(b) shows the results of the measure $\overline{Precision}$, when varying σ and d. Here, we can observe that the best choice of σ in the presented case is 0.1. Interestingly a further increase of σ leads to a dramatic loss in precision. This matches with the observation that can be made in Figure 4(d) and 4(h). Increasing σ results in a slow vanishing of FP and a faster vanishing of TP. Resulting from this, the output of precision decreases as well. Overall the effect of d is low in the considered case.

A similar effect can be seen when considering $\overline{Recall}$. Again the best results are achieved when using σ at around 0.1. The measure is computed using TP and TN. In Figure 4(l), we can observe that increasing σ results in less vanished values for TN. Therefore, this effects the $\overline{Recall}$ metric similarly to the $\overline{Precision}$ metric. In contrast to precision, for recall, the effect of d is high in the given case.

The effect of σ and d for $\overline{FPR}$ can be seen in Figure 9(d). When increasing σ, the result improves. This also holds for an increased d.

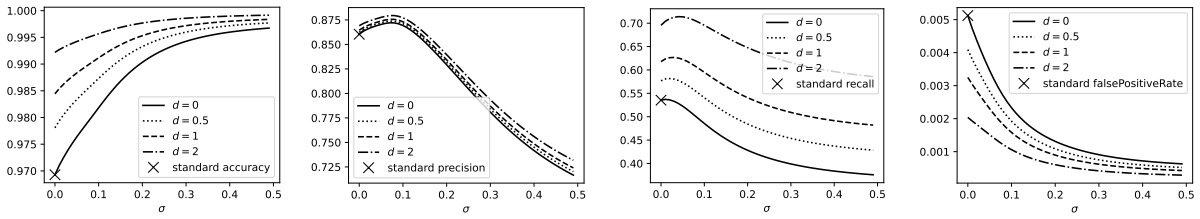
5 DISCUSSION

General Observations The presented metrics allow generalizing original machine learning performance metrics. When setting σ and d to 0, the resulting values are equal to the original computations.

The classifications that we proposed are based on a Gaussian distribution function but can be exchanged with any distribution function that holds an overall integral of 1. Also, the used damping function could be adapted if required. Here, functions that output 1, when a damping factor of 0 is used can be considered.

By using the adapted definitions of TP, FP, TN, and FN with a $\sigma > 0$ we can basically encode how far away the predictions are from the threshold. The benefit of this can be seen in Figure 5, wherewith increasing σ one can easily assess the quality of the threshold. For this particular example, it seems like the threshold is well chosen to classify the TN while keeping FN to a minimum. With increasing σ the TN stays the same, except at the boundaries, while the FN quickly fades away, which means that they are close to the threshold – they are classified with high uncertainty. The classifications of TP and FP also fade away relatively quickly. They are also relatively close to the threshold, and thus also relatively uncertain.

Using the damping factor allows to include the uncertainty captured in the made prediction of a machine



(a) $\overline{Accuracy}$ (b) $\overline{Precision}$ (c) $\overline{Recall}$ (d) $\overline{FPR}$

Figure 9: Results of the adapted metrics. Accuracy, Precision, Recall and FPR are affected by the choice of σ and d. The unmodified values of these metrics are indicated by a black x.

learning approach into the classification scheme. In Figure 7 it can be clearly seen that the classification around the general area of the lesion in the ground truth data is very uncertain. The same effect happens at the boundaries of the brain. Interestingly, the BNN predicts with high certainty FP. This is of course not desirable and such errors can be easily spotted with the method presented in this paper.

As a rule of thumb, it holds, that if for high σ and d the classifications of TP and TN are high and for the FP and negatives it is low, the model is trustworthy. This is also reflected in the adapted metrics, for example in the adapted accuracy metric depicted in Figure 9(a). The values are monotonically increasing with increasing σ and overall higher with higher d. One could infer, that our model is generally trustworthy where the uncertainty is low. If the graph in Figure 9(a) was monotonically decreasing, it would mean that the model predicts with high certainty wrong results, i.e. it is not that trustworthy. Care has to be taken for very unbalanced datasets, or datasets where one class can be much more easily identified than the other. This is the case for our model because a brain lesion can only occur in the brain, therefore a significant portion of the head scan can be easily classified as a TN with very high certainty. These problems can be alleviated by also considering the other metrics, like the F1-score.

Limitations Although the presented approach provides large flexibility, it also results in more input parameters. In this work, we showed that the influence of the input parameters can be inspected visually. Here, contrary to the original measurements, a visual inspection of the parameters is required.

In the presented work we showed that the provided measures are applicable for a BNN. We do not see limitations in the application to further networks, but we have not proven this statement.

6 CONCLUSION

This paper introduced adaptations to existing metrics for evaluating a binary classifier, that can incorporate uncertainty information from the model itself and uncertainty regarding the exact location of the threshold. For that, we use a Gaussian distribution function attached to the threshold and allow a damping factor for uncertainty-aware machine learning outputs. These metrics were applied to a real-world example of a Bayesian neural network to prove applicability.

As future work, we aim to use the measures in the back-propagation in the learning phase of neural networks. In addition, we further research the visual inspection of the chosen parameters of the presented measures.

7 REFERENCES

[1] A. I. Khan and S. Al-Habsi, "Machine learning in computer vision," *Procedia Computer Science*, vol. 167, pp. 1444–1451, 2020. International Conference on Computational Intelligence and Data Science.

[2] R. G. C. Maack, G. Scheuermann, H. Hagen, J. T. H. Penaloza, and C. Gillmann, "Uncertainty-aware Visual Analytics-Scope, Oppertunities and Challenges," PREPRINT (Version 1) available at Research Square, 2021.

[3] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim, "The role of uncertainty, awareness, and trust in visual analytics," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 240–249, 2015.

[4] E. Goan and C. Fookes, "Bayesian neural networks: An introduction and survey," *Lecture Notes in Mathematics*, p. 45–87, 2020.

[5] M. Kläs and A. M. Vollmer, *Uncertainty in Machine Learning Applications: A Practice-Driven Classification of Uncertainty: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings*, pp. 431–438. 01 2018.

[6] C. Gillmann, D. Saur, and G. Scheuermann, "How to deal with uncertainty in machine learning for medical imaging?," in *2021 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*, pp. 52–58, 2021.

[7] L. Sluijterman, E. Cator, and T. Heskes, "How to evaluate uncertainty estimates in machine learning for regression?," 2021.

[8] L. Nieradzik, G. Scheuermann, D. Saur, and C. Gillmann, "Effect of the output activation function on the probabilities and errors in medical image segmentation," 2021.

[9] Y. Ding, J. Liu, X. Xu, M. Huang, J. Zhuang, J. Xiong, and Y. Shi, "Uncertainty-aware training of neural networks for selective medical image segmentation," in *Proceedings of the Third Conference on Medical Imaging with Deep Learning* (T. Arbel, I. Ben Ayed, M. de Bruijne, M. Descoteaux, H. Lombaert, and C. Pal, eds.), vol. 121 of *Proceedings of Machine Learning Research*, pp. 156–173, PMLR, 06–08 Jul 2020.

[10] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12011–12020, 2020.

[11] R. Das, S. Sen, and U. Maulik, "A survey on fuzzy deep neural networks," *ACM Comput. Surv.*, vol. 53, may 2020.

[12] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods," *Machine Learning*, vol. 110, p. 457–506, Mar 2021.

[13] M. Sacco, J. Ruiz, M. Pulido, and P. Tandeo, "Evaluation of machine learning techniques for forecast uncertainty quantification," 11 2021.

[14] J. M. Twomey and A. E. Smith, "Performance measures, consistency, and power for artificial neural network models," *Mathematical and Computer Modelling*, vol. 21, pp. 243–258, 1995.

[15] O. Schoppe, N. S. Harper, B. D. B. Willmore, A. J. King, and J. W. H. Schnupp, "Measuring the performance of neural models," *Frontiers in Computational Neuroscience*, vol. 10, 2016.

[16] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," 10 2018.

[17] W. Imtiaz, H. Ghafoor, R. Sehar, T. Mahboob, and M. Khanam, "Evaluating the performance estimators via machine learning supervised learning algorithms for dataset threshold," *International Journal of Computer Applications*, vol. 119, pp. 1–6, 06 2015.

[18] S. Gao, W. Dong, K. Cheng, X. Yang, S. Zheng, and H. Yu, "Adaptive decision threshold-based extreme learning machine for classifying imbalanced multi-label data," *Neural Processing Letters*, vol. 52, pp. 1–23, 12 2020.

[19] V. Thada and V. Jaglan, "Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm," *International Journal of Innovations in Engineering and Technology*, vol. 2, pp. 202–205, 08 2013.

[20] S. Li, Y. Xie, and L. Song, "Data-driven threshold machine: Scan statistics, change-point detection, and extreme bandits," 2016.

[21] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: Analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, 08 2015.

[22] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis, "Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons," 2022.

[23] D. Conway and J. White, *Machine Learning for Hackers: Case Studies and Algorithms to Get You Started*. O'Reilly Media, 2012.

[24] C. K. Jones, G. Wang, V. S. Yedavalli, and H. I. Sair, "Quantifying epistemic and aleatoric uncertainty in 3d u-net segmentation," in *medRxiv*, 2021.

[25] K. Ramamurthy, R. Menaka, A. Johnson, and S. Anand, "Neuroimaging and deep learning for brain stroke detection - a review of recent advancements and future prospects," *Computer Methods and Programs in Biomedicine*, vol. 197, p. 105728, 08 2020.

[26] C. Gillmann, L. Peter, C. Schmidt, D. Saur, and G. Scheuermann, "Visualizing multimodal deep learning for lesion prediction," *IEEE Computer Graphics and Applications*, vol. 41, no. 5, pp. 90–98, 2021.

[27] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," 2017.